FEDERAL HIGHWAY ADMINISTRATION
CALIFORNIA DIVISION

Systems Engineering Guidebook For ITS

Version 1.1

February 14, 2005

Prepared by:

Siemens ITS
ASE Consulting LLC

In association with:
J & J Project Consultant
Jacoby Consulting

Acknowledgements

Sponsors of this Guidebook:

The sponsors of the Guidebook were the Federal Highway Administration and the California Department of Transportation's Division of Research and Innovation.

The Guidebook Development team:

Mr. Frank Cechini - Federal Highway Administration

Mr. Randy Woolley – California Department of Transportation

Mr. Hassan Ghotb – California Department of Transportation (Project Manager)

Mr. Warren Tighe – Siemens ITS (Contract Manager)

Mr. Michael E. Krueger - CSEP – ASE Consulting LLC (Technical Lead)

Mr. James Lewis – J and J Project Consultant

Dr. Carol Jacoby – Jacoby Consulting

Ms. Nancy Rantowich – SEA Consulting

Ms. Terri Parent – Technical Editor

The Review Team:

International Council on Systems Engineering (INCOSE)

Federal Highway Administration

California Department of Transportation:

- Division of Planning
- Division of Information Technology
- Division of Traffic Operations
- Division of Research and Innovation

Santa Clara Valley Transportation Authority

City of Santa Rosa

SYSTEMS ENGINEERING GUIDEBOOK FOR ITS

Table of Contents

Table of Figures

Table of Tables

Forward

Intelligent Transportation Systems (ITS) is now over 15 years old as a program of operational initiatives. Over this time, Intelligent Transportation *Systems* have gradually evolved, becoming more complex and integrated. It seems, though, that we are still in the infant stages of ITS developments. We have not seen the full benefit of how technology can make our transportation facilities more efficient. In many cases, we are still struggling with mainstreaming ITS into the traditional transportation planning and project development process. Several ITS programs have started with the best of intentions but have failed to produce their envisioned goals. Yet the vision of ITS is still alive and the need for these systems is greater now then ever and for good reason. The needs are dynamic and are constantly changing. Within the last 5 years, the concerns of conformity to air quality, congestion, and urban growth has dramatically expanded into security of people and facilities. For example, Amber Alert was not envisioned as part of the initial design of ITS, but now is a vital function of our transportation system. It utilizes the communications and changeable message signs to rescue children. The same is true for safeguarding our transportation infrastructure as cameras and communications are now being used to monitor, alert and respond to intruders.

The full expectation of sharing control and information among agencies, and the implementation of large regional multi-modal systems that integrate with the local agencies have yet to be fully realized. Institutional issues are still significant barriers in a number of regions; for example, agency contracting practices and policies regarding ownership of development products, managing operations and maintenance, and procurement practices. In large regional systems the question of who is going to pay for operations and maintenance continues to emerge.

What has also been a significant issue is the development of ITS projects. The reality of what stakeholders get from ITS developments often falls short of expectations they have at the start of the ITS project. Then there are schedule delays and cost overruns that have plagued ITS development.

We believe one of the key factors to many of the issues mentioned, is that no common process is in use for the development ITS systems, as there has been for traditional highway design. For this reason we believe this guidebook will benefit the ITS practitioner and reduce the risk of failed ITS projects and improve the interagency cooperation and coordination.

This Guidebook is intended to provide a set of system development process activities that have been used in the past in domains such as Information Technology, Department of Defense, Mil-Aerospace, NASA, and the automotive industry, where similar technologies are used. Most of the principles and processes in this Guidebook are very similar and in some cases identical to these in other domains but with a focus on the unique aspects of the ITS domain. Processes have been added or modified for the Intelligent Transportation Systems developments where appropriate.

The ITS practitioner using this Guidebook will need to tailor the process activities for the size, risk, and complexity of each project. The Guidebook provides guidance in tailoring for each process activity and provides three example projects for guidance.

Our expectation is that this Guidebook will provide you, the ITS practitioner, a set of tools that will support you in the development of your ITS project.

The Authors of this Guidebook are anxious to hear about your experience in using this Guidebook in the development of your ITS project.

Best Regards

Authors of the Systems Engineering Guidebook for ITS

Feedback from You!

We want this Systems Engineering Guidebook to be of value to you!

In order to validate what we recommend in this Guidebook, we need to hear from you about your "real world experience" in the use of the Guidebook. This will allow us to continually improve each of the process activities and to update the Guidebook as appropriate.

Please let use know what you think when applying the principles, recommendations, tips, and checklists of this Guidebook. Please let us know how it works for you!!

Please send all comments and Feedback to:

Subject line: SE Guidebook for ITS Comments

Randy.Woolley@dot.ca.gov or

Frank.Cechini@fhwa.dot.gov or

Michael.krueger@ase-consult.com

1 Executive Overview

In the late 1980's, the transportation community envisioned Intelligent Transportation Systems (ITS) to be a tool for transportation practitioners to make transportation facilities more efficient and to encourage a more regional view of transportation. What was not well understood at the time was the extent of new sets of skills, capabilities, and interagency cooperation that the transportation agencies would need to meet these goals. There also was not recognition of the importance of addressing lifecycle operations and maintenance. Now, there is an awareness of these key ITS challenges. To address them, in the mid to late 1990's, systems engineering was introduced to the ITS community and it resonated with a number of ITS practitioners. As a result, the FHWA issued 23 CFR 940 that requires all ITS projects funded with highway trust funds to be based on a systems engineering analysis.

The goal of this Guidebook is to help agencies use common, consistent and well-established systems engineering tools and processes to:

- **Improve the quality of Intelligent Transportation Systems**

Systems engineering thinking promotes increased up-front planning and system definition prior to technology identification and implementation. Documenting stakeholder needs, expectations, the way the system is to operate (Concept of Operations), and the system requirements (WHAT the system is to do) prior to implementation leads to improved system quality.

- **Reduce the risk of cost and schedule overruns**

Systems engineering promotes stakeholder involvement throughout project development and improves project control with clearly defined decision points (Control Gates). With the up-front planning described above, the risk of costly rework and schedule slips during the latter stages of implementation are greatly reduced.

- **Gain participation of multiple agencies and a diverse set of stakeholders**

Participation of stakeholders is essential for successful system developments. Using a common and standard development process enables stakeholders to understand and actively participate in the development, as well as reducing the learning curve when new stakeholders get involved in a project. A common process ensures a wider set of resources (staff, expertise) that agencies can draw upon within the project lifecycle.

- **Maintain, operate and evolve the Intelligent Transportation System**

Project developments that use a systems engineering approach will improve the documentation of the system (specifically requirements, design, verification, development and support documentation). Having this documentation will improve the long-term operations, maintenance, upgrades and expansion capabilities of the system.

- **Maintain consistency with the regional and state ITS architectures**

Once a regional ITS architecture is developed and projects are defined, a common and clear roadmap for ITS project development is laid out. The systems engineering approach enables consistency with the regional ITS architecture to be verified and maintained.

- **Provide flexibility in procurement options for the agencies**

Intelligent Transportation Systems that are well documented will have greater flexibility in procurement options. Proprietary developments are minimized, proprietary subsystems are identified and the use of industry standard interfaces promoted. This enables alternate system integrators and consultants to support the agencies in upgrades and system expansion. In other words, it minimizes the agencies' need to be "locked into" a specific vendor or system integrator.

- **Keep current with rapid evolution in technology and needs of transportation**

One of the challenges for agencies is to stay current with the rapid changes in technology. Intelligent Transportation Systems are long term investments for agencies and it is important to avoid technology obsolescence. In other words, when field devices fail, the agency should be able to replace them without a major development effort and without maintaining large inventories. Systems engineering promotes system modularity and the use of standard interfaces where possible. If a technology changes or is no longer available, the functionality can be replaced with minimal impact to other parts of the system (goal of plug and play).

For whom was this Guidebook designed?

This Guidebook's primary audience is the agencies that plan, implement, manage and operate Intelligent Transportation Systems and, within the agency, management that champions ITS projects and supports the agency's ITS practitioners. In addition, this Guidebook will help consultants and system integrators, who would be potential contractors for the agencies, gain an understanding of the required systems engineering processes. This Guidebook identifies roles and responsibilities for project development and provides a common process and language so that agencies, system integrators, and consultants can have the same understanding as to what is to be expected when developing ITS projects.

How should this Guidebook be used and what is in it?

This Guidebook is a reference to help practitioners develop Requests for Proposals, assess capabilities of potential Systems Managers (Systems Engineering Technical Assistance, and Independent Verification and Validation consultants) and development teams (System Integrators), and to help guide the ITS practitioner through the development of ITS projects.

The Guidebook provides guidance for the following: (this list is not all inclusive)

- Lifecycle phases for Intelligent Transportation Systems
- Activities needed to carry out each development task (based on industry best practices)
- Tailoring development activities to fit large and small projects (tailoring up and tailoring down respectively)
- Roles and responsibilities in project development
- Important activities that the system owner needs to be involved with
- Activities to ensure that all the bases are covered for each activity
- Tips, cautions and other essential information needed for a task
- Applicable industry standards
- Format and information needed for the development of key project documents

What does the Guidebook NOT cover?

This Guidebook was not intended to be an in-depth textbook on systems engineering. Section 7 has reference material that will direct the reader to a number of books, papers and standards that are on the market and provide excellent material to augment this Guidebook. This Guidebook does not provide guidance for the development of regional architectures. This is well covered in "Regional ITS Architecture Guidance: Developing, Using & Maintaining an ITS Architecture for Your Regions" prepared by the National Architecture development team, October 12, 2001.

How is this Guidebook organized?

Figure 1-1 illustrates the organization of the Guidebook. The outer layer is the Executive Summary providing an overview of the Guidebook, and the next layer is a closer look at the systems engineering environment. Then the steps of processes and crosscutting activities are described followed by the foundation of roles and responsibilities, capabilities needed, and finally example reference and supporting materials.

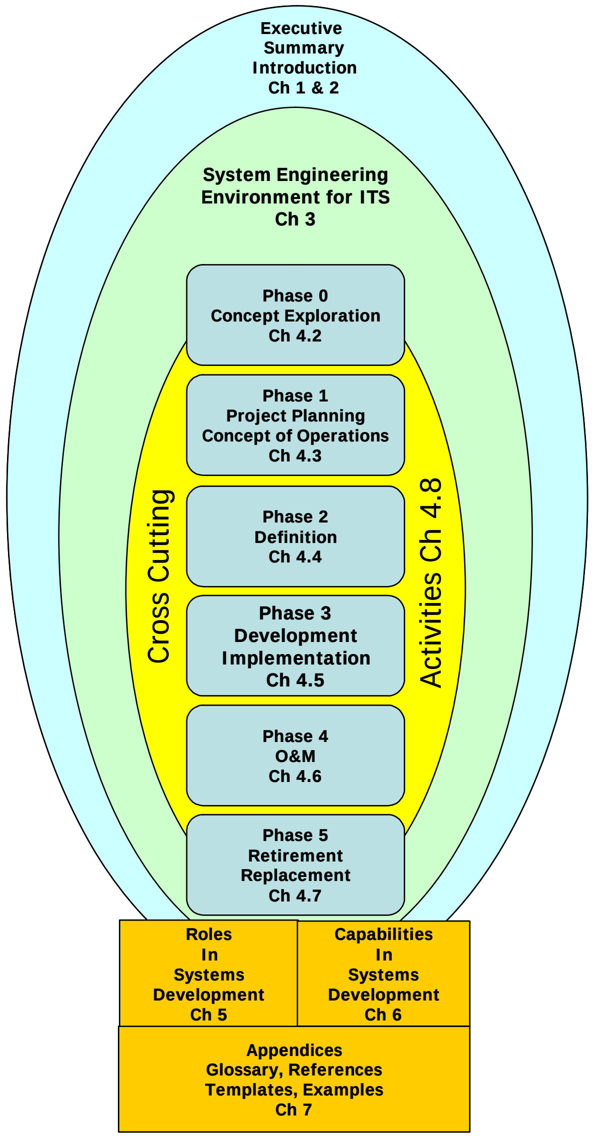


Figure 1-1 Organization of the SE Guidebook

Understand the Guidebook and the systems engineering process (Sections 1 and 2). The first step is to understand the organization of the Guidebook, and the necessary steps of the systems engineering process. These chapters will point the reader to the relevant overview sections. Chapter 1 is the Executive Summary, which gives a short overview of the entire Guidebook. This is intended for managers or others who wish a quick view of the processes and key concepts presented here. Chapter 2 places the Guidebook into context in terms of purpose and scope.

Analyze and prepare the systems engineering environment (Section 3). There are many factors that both support and constrain the systems engineering process for ITS. The Guidebook user needs to be familiar with these factors before starting work. Examples are the National ITS Architecture, FHWA Final Rule, ITS standards, and agency roles and responsibilities. This chapter also guides you in tailoring the systems engineering process to fit your particular project.

Form your team (Sections 5 & 6). These chapters discuss the typical roles and capabilities of agencies, consultants, and developers. While such roles vary greatly from agency to agency, this will give guidance in putting together a project team.

Initiate crosscutting activities (Section 4.8). There are several important activities that are ongoing continually or repeatedly throughout the systems engineering process. These include elicitation, project management, acquisition planning, generation of deliverables and documentation, process improvement, configuration management interface management, risk management, program metrics, control gates, trade studies, technical reviews, and stakeholder involvement. These activities support the tasks carried out during the six phases.

Follow the systems engineering process (Section 4.1 – 4.7). This is the heart of the Guidebook. The process follows six phases, as shown in the center of the diagram. Section 4.1 provides an overview, diagrammatic roadmap, and links to the key discussions in section 4. The other sections each correspond to the major phases of project development: concept exploration and feasibility assessment, project planning and concept of operations, system definition, system development and implementation, operations and maintenance, and retirement/replacement. A control gate that must be passed in order to proceed follows each phase.

1.1 Overview of the Vee Technical Development

The Vee Development Model is the recommended development model for ITS projects. This model for systems development combines the important features of the classic Waterfall model and the Spiral development model used primarily for software development. Both of these models are briefly described below.

Illustrated in Figure 1-2 is the Vee Development Model in context of the lifecycle framework. This model has gained much acceptance in the systems engineering community and has been illustrated as part of the most recent Systems Engineering Process Standards ISO 15288 and EIA 632 as well as many of the current leading systems engineering texts. The reason for this acceptance is that the model illustrates some key systems principles about the relationship of the early phases of the development to the end results of the project. This is described in more detail in the step-by-step description below. This overview also serves as a primer for the reader who is not familiar with the systems development process.

The following are step-by-step descriptions of the lifecycle model and the crosscutting activities that support the steps of the lifecycle. The title of each section is followed by the number of the section in this Guidebook which contains more descriptive detail. In addition to this description, observations about the Vee Development Model, some basic systems engineering principles, terms and definitions are discussed to give the reader a starting point with this section of the Guidebook. A more comprehensive list of terms and definitions are included in the appendix. The Vee portion of the illustration is the project level development phase. This discussion starts with the description of the left "wing" of the illustration, the Vee technical model itself and finally the right "wing" of the lifecycle framework. It should be noted that the "Changes and Upgrades" step (right "wing") is performed using the Vee technical model but is not illustrated that way for the purposes described below.

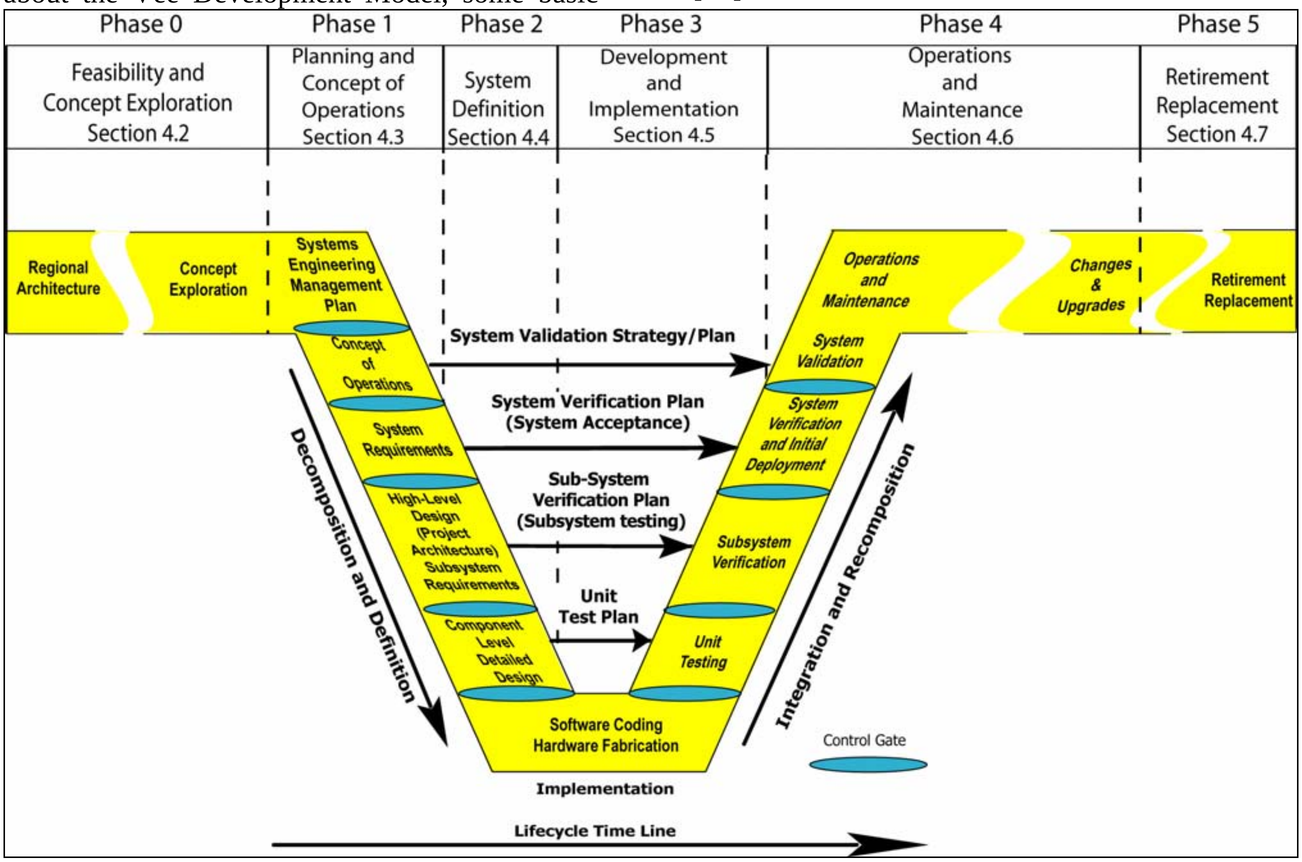


Figure 1-2 ITS Project Lifecycle Phases and the Lifecycle Tasks in this Guidebook

Basic Terms and Definitions

Architecture: *Two definitions –*

1) **Regional -** a framework for ensuring technical and institutional integration of ITS systems in a geographic area. For these purposes, a regional (ITS) architecture is based upon the National ITS Architecture tool; 2) **Project -** a project-specific description of both logical and physical elements arranged in a hierarchical form and shows interconnections among the elements. It has enough definition that component level detailed design specification can be written and developed.

System - is an integrated composite of people, products and processes, which provide a capability to satisfy a stated need or objective.

Systems Engineering - is an inter-disciplinary approach and a means to enable the realization of successful systems. Systems engineering requires a broad knowledge, a mindset that keeps the big picture in mind, a facilitator and a skilled conductor of a team.

FHWA Final Rule **-** The FHWA Rule on Architecture Standards and Conformity (Final Rule), also referred to as 23 CFR 940, requires the development of regional ITS architectures (RA's) and that all ITS projects using Federal funds be developed using a systems engineering analysis. The systems engineering analysis includes: identification of the portion of the RA being implemented, participating agencies roles and responsibilities, requirements definition, alternatives analysis, procurement options, identification of applicable ITS standards and testing procedures, and procedures and resources for system operations and management.

Process Activities

Section 4.2.1 to 4.7.1

The following is a summary of the process steps in the Vee technical model.

Interfacing with the regional ITS architecture and planning (4.2.1)

This initial step interfaces with the ITS architecture for a region. Development of a regional ITS architecture is not covered by this Guidebook and is well described in "ITS Regional Architecture Guidance Document: Developing, using and maintaining an ITS architecture for your region – October, 2001. Two of the key activities of this phase are the identification of the regional stakeholders and the building of consensus for the purposes of information sharing and long term operations and maintenance. The candidate projects are then put into the transportation planning process (TIP, STIP, RTIP). For more information on developing a regional ITS architecture please refer to Regional ITS Architecture Guidance document from the website: http://www.its.dot.gov/arch/arch.htm.

Concept Exploration and Feasibility Assessment (4.2.2 & 4.2.3)

Concept Exploration is used to perform an initial feasibility and benefits analysis and needs assessment for the candidate projects from the regional ITS architecture development. This would result in a feasibility study report and specific cost benefit analyses for alternative project concepts. The output of this stage is a definition of the problem space, key technical metrics and refinements to the needs, goals, objectives and vision. The stage identifies the highest cost/benefit project concept (best business case); the one that should move forward into development. This activity may result in combining or dividing candidate projects based on the best cost/benefit analysis.

Systems Engineering Planning (4.3.1 & 4.3.2)

Each project that moves forward into development must be planned. This planning takes place in two parts. In part one, the system owner develops a set of master plans and schedules that identifies what plans are needed and, at a high level, the schedule for the implementation of the project. This becomes the framework for what is developed in part two. In part two, the plans are completed during the steps from the concept of operations to the high level design. These plans, once approved by the system owner, become the control documents for completion of the development and implementation of the project.

Concept of Operations (4.3.3)

The Concept of Operations is the initial definition of the system. At this stage, the project team documents the way the envisioned system is to operate and how the envisioned system will meet the needs and expectations of the stakeholders. The envisioned operation is defined from multiple viewpoints for example, operators, maintainers, and managers, and how the system will be validated (proof that the envisioned system meets the intended needs). A refinement of the problem space definition, needs, goals, expectations, stakeholder lists, and project constraints is placed into the concept of operations document. This document contains the updated, refined summary of the work done at the concept exploration phase.

System Level Requirements (4.4.1)

Requirements are developed for the system. At the system level, the definition of WHAT the system is to do, HOW WELL it is to do it and under WHAT CONDITIONS are documented. The system requirements are based on the user needs from the Concept of Operations. Requirements do not state HOW (design statements) the system will be implemented unless it is intended to constrain the development team to a specific solution.

High Level Design (Project Architecture) and Sub-system Requirements (4.4.1 and 4.4.2)

The high level design stage defines the project level architecture for the system. The system level requirements are further refined and allocated (assigned) to sub-systems of hardware, software, databases and people.

Requirements for each sub-system element are documented the same way as was done for the system level requirements. This process is repeated until the system is fully defined and decomposed. Each layer will have its own set of interfaces defined. Each layer will require an integration step that is needed when the sub-system is developed. The control gate that is used for this final review is sometimes called the Preliminary Design Review (PDR).

Component Level Detailed Design (4.4.3)

At the component level detailed design step, the development team is defining HOW the system will be built. Each sub-system has been decomposed into components of hardware, software, database elements, firmware and/or processes. For these components, detailed design specialists in the respective fields create documentation ("build-to" specifications) that will be used to build or procure the individual components. A final check is done on the "build–to" specifications before the design moves forward to the actual coding and hardware fabrication. At this level the specific commercial off-the-shelf hardware and software products are specified but they are not purchased until the review is completed and approved by the system owner and stakeholders. The control gate that is used for this final review is sometimes called the Critical Design Review (CDR).

Hardware/Software Procurement or Development (4.5.1)

This stage involves hardware fabrication, software coding, database implementation and procurement and configuration of off-the-shelf products. This stage is primarily the work of the development team. The system owner and stakeholders monitor this process with planned periodic reviews, e.g. code walkthroughs and technical review meetings. Concurrent with this effort, unit test procedures are developed that will be used to demonstrate how the products will meet the detailed design. At the completion of this stage the developed products are ready for unit test.

Unit Testing (4.5.3)

The components from the hardware and software development are verified in accordance with the unit Verification Plan. The purpose of unit test is to verify that the delivered components match the documented component level detailed design. This is done by the development team in preparation for the next level of integration. This is a good review point for the system owner and stakeholders.

Sub-system Integration and Verification (4.5.2, 4.5.3)

At this step, the components are integrated and verified at the lowest level of the sub-systems. The first level of verification is done in accordance with the Verification Plan and is carried out in accordance with the Verification Procedures (step-by-step method for carrying out the verification) developed in this stage. Prior to the actual verification a test readiness review is held to determine the readiness of the sub-systems for verification. When it is determined that verification can proceed, the sub-systems are then verified. When the integration and verification is completed, the next level of sub-system is integrated and verified in the same manner. The process continues until all of the sub-systems are finally integrated and verified.

System Verification (4.5.3)

System verification is done in two parts, the first part is done under a controlled environment (sometimes this is called a "factory test") and the second part is done in the environment in which the system is intended to operate (sometimes called "on-site testing") and is done after initial system deployment. At this stage the system is verified in accordance with the Verification Plan developed as part of the system level requirements done earlier in the development. The system acceptance will continue through the next stage, initial system deployment. The final part of system verification is then completed. A control gate is used for this conditional system acceptance.

Initial System Deployment (4.5.4)

At Initial System Deployment, the system is finally integrated into its intended operational environment. This step may take several weeks to complete to ensure that the system operates satisfactorily long term; this is sometimes called a "system burn-in". Many system issues will surface when the system is operating in the real world environment for an extended period of time. This is due to the uncontrollable nature of inputs to the system, long term "memory" leaks in software coding and race conditions. (Unexpected delays between signals) that may only occur under specific and infrequent conditions. Once the system verification is completed, the system is accepted by the system owner and stakeholders and moves into system validation and operations and maintenance phases.

System Validation (4.6.1)

Validating the system is a key activity of the system owner and stakeholders. It is here that they will assess the system's performance against the intended need, goals and expectations as documented in the Concept of Operations and in the Validation Plan. It is important that this validation takes place as early as possible after the acceptance of the system in order to assess the strengths and weaknesses and assess new opportunities. As a result of the validation new needs and requirements may result. This activity does not check on the work of the system integrator or component supplier (that is the role of System Verification) and is performed after the system has been accepted and paid for. As a result of validation new needs and requirements may be identified. This evaluation sets the stage for the next evolution of the system.

Operations and Maintenance (4.6.2)

After the initial deployment and system acceptance, the system moves into the operations and maintenance phase. In this phase, the system will carry out the intended operations for which it was designed. During this phase, routine maintenance is performed as well as staff training. This phase is the longest phase since it will extend through the evolution of the system and end when the system is retired or replaced. This phase may carry on for decades. It is important that there are adequate resources to carry out the needed operations and maintenance activities; otherwise, the life of the system can be significantly shortened due to neglect.

Changes and Upgrades (4.6.3)

During the operations and maintenance phase, if changes and upgrades are needed, it should be done in accordance with the Vee technical process as recommended by this Guidebook. Using the Vee process for changes and upgrades will help maintain system integrity (maintain synchronization between the system components and its respective documentation). Sometimes existing systems (legacy systems) have not been well documented. In such cases, it is recommended to first perform a reverse engineering process on the target areas of proposed change in order to develop the needed documentation for the forward engineering process.

Retirement/Replacement (4.7.1)

At some point in the life of a system, it may be necessary to retire and/or replace the system. The system may no longer be needed, may not be cost effective to operate, may no longer be maintainable due to obsolescence of key system elements or this may be a planned activity where an interim system was put in place for a period of time until the final system was ready for deployment. This stage looks at how to monitor, make the decisions needed and prepare for this event.

Cross-Cutting Activities
Sections 4.8.1 to 4.8.12

A number of cross cutting activities are needed to support the development of Intelligent Transportation Systems. The following are the essential enabling activities used to support one or more of the life-cycle process steps.

Stakeholder Involvement (4.8.1)

Stakeholder involvement is regarded as one of the most critical enablers within the development and life-cycle of the project and system. Without effective stakeholder involvement, the systems engineering and development team will not gain the insight needed to understand the key issues and needs of the system owner and stakeholders. This will increase the risk of not getting a valid set of requirements to build the system or to get buy-in on changes and upgrades.

Elicitation (4.8.2)

Elicitation is the act of effectively and accurately gathering information needed to develop the system. Needs, goals, objectives, requirements, and other information are obtained by a discovery process. Some of the information is documented or otherwise clearly stated but much is implied or assumed. This enabling process helps draw out and resolve conflicting information, build consensus, document and validate this information.

Project Management Practices (4.8.3)

Various project management practices are needed to support the development of the system. Project management practices provide a supportive environment for the various development activities. It provides the needed resources, then monitors and controls cost, schedules and communicates status between and across the development team members, system owner and stakeholders.

Risk Management (4.8.4)

There will be risks for ITS system development efforts. Risk Management is a process used to identify, analyze, plan, monitor and then to mitigate, avoid, transfer or accept these risks.

Project Metrics (4.8.5)

Project metrics are measures that both the project manager and the systems engineer use to track and monitor the project and the expected technical performance of the systems development effort. The identification and monitoring of metrics are important so that the team can determine if the project is "on-track" both programmatically and technically.

Configuration Management (4.8.6)

Managing change to the system is a key process that occurs throughout the life of the system. Configuration management is the process that supports the establishment of system integrity (the documentation matches the functional and physical attributes of the system) and maintains this integrity throughout the life of the system (synchronizes changes to the system with its documentation). The lack of change management will shorten the life of the system and may prevent a system from being implemented and deployed.

Procurement Options (4.8.7)

Procurement options are important for the system owner and stakeholders. The goal in choosing a procurement option is to give the system owner the greatest flexibility and to manage project risk appropriately. The choice depends on the phase of work being done. Some phases of work will lend themselves better to one type of procurement option over another.

Deliverables/Documentation (4.8.8)

Examples of products are identified as one would expect from each phase of the development and system lifecycle. Asking for the appropriate documentation at the appropriate level of quality will drive the quality of system that will be delivered.

Process Improvement (4.8.9)

A quality aspect of the systems lifecycle is to continuously improve the process and to learn from previous efforts to improve future work that may be done. Process improvement is an enabler that will provide insight on what worked and what needs improvement in the processes. This activity is used to improve the system owner's and development team's documented processes over time.

Control Gates (4.8.10)

Control Gates are formal decision points along the lifecycle that are used by the system owner and stakeholders to determine if the current phase of work has been completed and that the team is ready to move into the next phase of the lifecycle. By setting entrance and exit criteria for each phase of work, the control gates are used to review and accept the work products done for the current phase of work and also evaluate the readiness for moving to the next phase of the project.

Trade Studies (4.8.11)

Technical decisions on alternative solutions are a key enabler for each phase of system development. This starts when alternative concepts are evaluated, and continues as requirements are decomposed and allocated to sub-system developing, the high level design is developed and commercial off the shelf products are assessed. This section provides a method to perform a trade study.

Technical Reviews (4.8.12)

Technical reviews are used to assess the completeness of a product, identify defects in work, and align the team members to a common technical direction. This section provides a process for conducting a technical review.

Key Observations for the Vee Development Model

1. The left side is the definition and decomposition of the system into components that can be built or procured. The bottom of the Vee is the construction, fabrication and procurement or development of the component items. The right side of the Vee integrates the components into sub-systems and finally into the final system. Each level of integration is verified against the left side of the Vee through the Verification Plans (verification process (4.5.3)).
2. Control gates (4.8.10) provide the system owner with formal decision points to proceed to the next step of the process. A control gate is an interface from one phase of the project to the next and there is an interface between each phase on the left side to the right side.
3. There is a relationship of the activities performed on the left side of the Vee to the products produced, integrated and verified on the right side of the Vee (model versus reality).

4. The view of the system that is most important for the system owner and stakeholders is at the Concept of Operations level. Below this level is the area of most interest to the development team and the area for which they are responsible (system owner responsibility versus the development team responsibility).
5. Importance of stakeholder involvement shows on the left side for defining the system and on the right side for the verification of the system.

1.2 Questions that this Guidebook Addresses

Is systems engineering just an elaborate process that will unduly burden the ITS practitioner?

No. When applied correctly, systems engineering requires more effort at the beginning of the project but will reduce effort in re-work during and at the end of the project with an overall schedule savings.

Systems engineering is associated with a set of processes, and, if it is viewed **only** as a series of required activities without consideration of the complexity of the system, it can become a burden on your project. **This is not the intent of systems engineering or this Guidebook. Systems engineering is also a mind-set called "systems thinking".** The challenge is to use systems thinking to tailor these processes into a set of activities that will successfully develop and deliver Intelligent Transportation Systems in the most efficient way.

The following are a few examples of systems engineering principles that express “systems thinking” and are needed to tailor the process according to the project complexity:

- Understand the problem to be addressed first
- View the problem and solution from the stakeholder points of view – walk in the shoes of the system owner and stakeholders
- Start at the finish line – define the output of the system and the way the system is going to operate
- Address project risks as early as possible, when the cost impacts are lowest
- Push technology choices to the last possible moment – define what is to be done before defining how it is to be done (form follows function)
- Focus on interfaces of the system and of the project (organizational, teams and process interfaces)
- Understand the organization of the system owner and stakeholders to enable stakeholder participation

This Guidebook is not intended to be “one size fits all” for systems development. It is important to assess each ITS project for risk and quality, and to define clear outcomes rather than to follow a “script”. Applying systems thinking to a project is essential to the tailoring of the processes to achieve the required level of system quality. This Guidebook will provide the best practices when applying the steps of systems engineering process.

Are there any benefits gained by doing systems engineering on my projects?

Yes. The primary benefit of doing systems engineering is that it will reduce the risk of schedule and cost overruns and will provide a system of higher integrity. Other benefits include:

- Better system documentation
- Higher level of stakeholder participation
- System functionality that meets stakeholders’ expectation
- Shorter project cycles
- Systems that can evolve with a minimum of redesign and cost
- Higher level of system reuse
- More predictable outcomes from projects

There are many studies that have been done that show the importance of using systems engineering principles. These reports document that using systems engineering principles has reduced the risks of project overruns and schedule delays when applied correctly. (See the following references in section 7 1) Standish research group study – Chaos 1994 and updated in 2000, 2) NASA studies, and 3) the INCOSE center of excellence)

Is systems engineering right for me, especially on my small projects?

Yes! Systems engineering should be applied on all projects small or large, simple or complex. The degree of formality and rigor that one applies to the systems engineering process will vary depending on the complexity of the project. This is called tailoring. All projects need to be assessed as to the amount of formal systems engineering process needed.

Systems engineering thinking is critical on all ITS projects. The systems engineering processes and techniques support systems thinking. The systems engineering processes and techniques must be scaled and tailored appropriately to all ITS projects. This Guidebook gives guidance on tailoring for each step of the process and recommendations based on example projects.

The tailoring needed for a project depends on the following project risk factors:

- System and institutional complexity (institutional issues, interfaces, technology)
- Number and type of stakeholders (integration of transportation and/or non-transportation agencies, scale of project)
- Inter-agency decisions and agreements that need to be made (sharing of control and data)
- Existing and needed documentation for the evolution of the system (legacy and new systems documentation for maintenance, expansion, and replacement)

Can I leverage existing agency resources to help me with systems engineering on my project?

Yes. The extent of this leveraging will depend on the size of and the expertise within the agency and/or cooperative agreements with other agencies, e.g. MPO, State DOT, adjoining public agency, federal resources, and systems engineering consulting services.

In organizations, often there are existing capabilities, processes, tools, and products that can be leveraged for the systems engineering support environment. For example, products from the training, information technology, asset management, quality assurance, risk management, and legal organizations can be used as a starting point for ITS projects.

This Guidebook describes the roles, responsibilities, and activities of the system owner, systems manager (Systems Engineering Technical Assistance, Independent Verification and Validation), and the development team (Systems Integrator) throughout the project lifecycle. These activities may be performed by agency personnel, contracted, or by some combination of the two. However, there are certain activities that are important for the system owner or his designate to perform. These activities are identified in each step of the systems engineering process in this Guidebook.

Can the systems engineering processes fit into the transportation project development cycle?

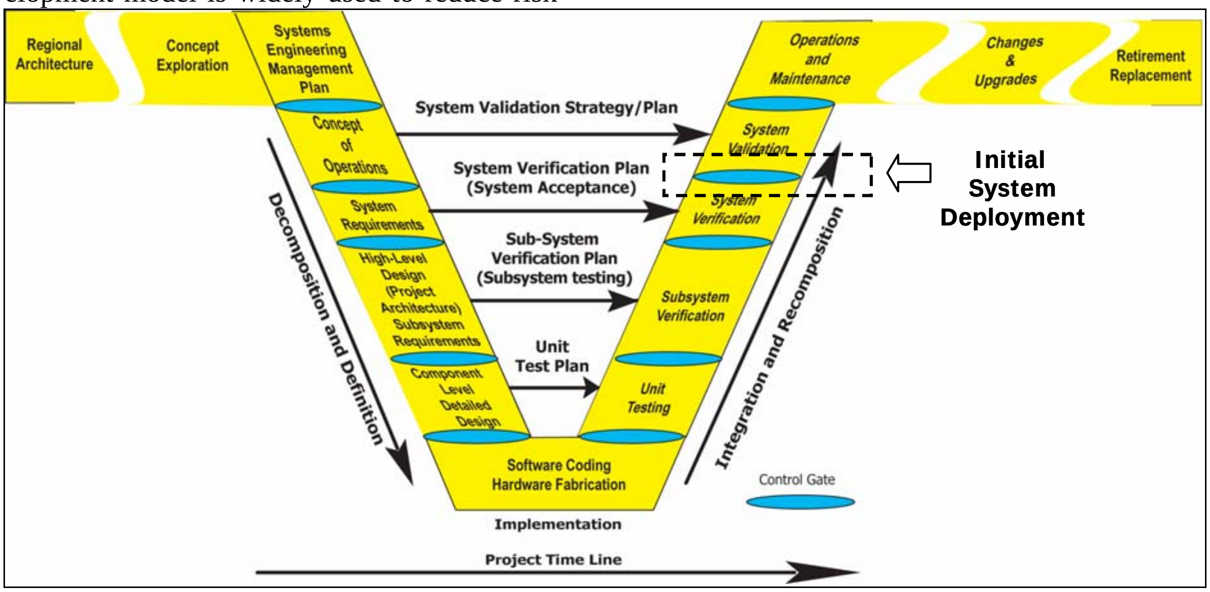
Yes. The systems engineering process is not new to the transportation domain. A systems approach has been used to build capital projects (highways) for many years, and the “systems thinking” identified earlier has been applied. The basic phases used for transportation development projects (study, concept exploration, definition, implementation, operations and maintenance and rehab/replace) are also the same phases used for high technology projects like Intelligent Transportation Systems (ITS) projects. What is unique to ITS is the rapidly changing technology in communications, hardware and software, and the iterative nature of designing and implementing software systems, along with their associated tools. As a result, the tasks and activities of the systems engineering process are different for ITS to accommodate this reality.

Are there different systems engineering development models that can be used for Intelligent Transportation Systems and which one is the best?

Yes. The classic system development models include: Waterfall, Spiral and the Vee development models. This Guidebook describes the various systems development models and delivery strategies with examples of types of ITS projects for each. The Vee Development Model is used as the overall framework for the project cycle as illustrated in Figure 1-3 below. The Guidebook uses the Vee model (tailored for ITS) originally developed by NASA in the late 1980’s for software development and then adapted for system development by Kevin Forsberg and Hal Mooz in 1990. The Vee Development Model is the third generation of models that integrates the original Waterfall and the Spiral models. The Vee Development Model is recommended by the Federal Highway Administration as the preferred method for Intelligent Transportation Systems development and is taught by the National Highway Institute. Today, the Vee Development Model is part of systems engineering standards like EIA 632 and ISO 15288. It has become popular in a number of domains, like automotive, banking, as well as defense and aerospace. The reason for its popularity is that it illustrates some key systems engineering principles that were not illustrated in the other two models, such as:

- The relationships of the outputs from early phases of the project to the later phases of the project
- The illustration of control or decision gates
- Involvement of the stakeholders in the early phases of the project

The other models have a role in systems development. For example, the spiral development model is widely used to reduce risk for some aspects of software development such as graphical user interfaces and algorithm development for processing information. When used in context of the Vee Development Model, the spiral can be used in the individual phases before proceeding to the next phase.



(Forsberg, Mooz, Cotterman INCOSE 1992)

Figure 1-3 Adapted from the Vee Technical Development Model

1.3 Summary

This Guidebook:

- Provides the agencies a resource that will help improve the development of Intelligent Transportation Systems
- Defines a common project process for multi-agency ITS projects improving development, coordination, participation, operations and maintenance, and integration
- Provides guidance for consultants and system integrators to meet an agency's expectations for the development process of ITS systems

This Guidebook, along with training, will help promote the use of systems engineering in ITS, and, as these processes become integrated into the transportation project development processes, will add yet another set of tools that ITS practitioners can use to improve transportation facilities.

Under-girding the Guidebook is a set of systems engineering principles that reach outside ITS projects, providing value to capital developments, research and information technology projects.

A common and defined process will enable a broad set of resources to contribute to agency projects (similar to that currently done as part of capital projects, enabling the plans and specifications to be developed anywhere in the state, making use of available resources as needed). Expertise in ITS will be broadened in the same way, and a pool of resources will be available to support ITS projects for any statewide agency.

2 Introduction

Intelligent Transportation Systems in the late 1980's was envisioned to be a tool for transportation practitioners to make transportation facilities more efficient and to encourage a more regional view of transportation. What was probably not well understood at the time was the extent of new skills and capabilities that the transportation agencies would need to implement and meet the goals of ITS. There is now an awareness that implementing ITS is more challenging than expected. In the mid to late 1990's, systems engineering was introduced to the ITS community and it resonated with ITS practitioners. In 2001 the USDOT issued a new regulation that requires a systems engineering approach to the implementation of ITS projects. With the further recognition that additional guidance was needed, this Guidebook was conceived. This Guidebook has the following seven major sections:

Section 1 is an executive overview

Section 2 is the introduction, purpose, scope, background, intended audience, how to use the Guidebook, and a brief introduction to the systems engineering lifecycle phases, key milestones, and activities.

Section 3 describes the systems engineering environment, estimating the amount of systems engineering needed, factors that drive the systems engineering environment, development models and strategies, relationship to the National ITS Architecture, transportation planning and standards. Also included in section 3 is what is needed in the system owner support environment, common system owner activities that already exist, and an introduction to systems engineering organizations.

Section 4 is the core of this Guidebook and describes the systems engineering process from interfacing with the regional ITS architecture to replacement and retirement.

Section 5 describes the roles and responsibilities for the system owner, consultant and the development team independent of whether it is in-house or contracted.

Section 6 describes the capabilities that are standard in the industry for systems engineering and looks briefly at the Capabilities Maturity Model Integrated (CMMI) which is a standard way to assess how well systems engineering is performed.

Section 7 contains appendices of useful systems engineering documentation guidelines and reference material.

2.1 Purpose

The following lists the purposes for this Guidebook:

Provide state and local agencies assistance, guidance and a standardized systems engineering approach to the development of ITS projects.

Provide a guide to industry best practices in systems engineering and lessons learned from other domains and past experience.

Provide guidance on compliance with the FHWA Final Rule (23 CFR 940 Part 11) pertaining to systems engineering analysis for the implementation of ITS projects.

This Guidebook is intended to be a guide to applying systems engineering practices and principles to the acquisition of Intelligent Transportation Systems and oversight in ITS developments.

2.2 Scope

This Guidebook covers the ITS project lifecycle, starting with interfacing to the portion of the regional ITS architecture to be implemented, and continuing through system retirement and replacement. This Guidebook does not cover how to develop and maintain a regional ITS architecture nor is it an in-depth systems engineering handbook. This Guidebook will address the interface between the planning and implementation of the projects and the interface between implementation of the project and the operations and maintenance of the system, and all the steps in between.

This guide identifies the expected outcomes for each step of the systems engineering process and identifies the roles and responsibilities for the system owner, systems engineering assistance (consultants) and the development team. Each process step will be described using a range of guidebook aids such as, "checklists", "tips", "process charts", examples and document guidelines in the appendix portion of this Guidebook. This Guidebook is not intended to be a comprehensive handbook on systems engineering. It is intended to provide an overview of the systems engineering process and its supporting and crosscutting activities.

The intent is to give owning agencies enough understanding of the systems engineering process to work with contractors and to understand what the contractors are providing and why, and to support staff and in-house activities. It will clarify and support their own role in the process as managers of contractors, employees, and the systems engineering activities. It also provides guidance and pointers to resources for systems engineering performed in house.

2.3 Background

The systems engineering process is not new to the transportation domain. A systems approach has been used to build capital projects (highways) for many years. What is relatively new is the application of rapidly changing technology to the transportation domain called Traffic Operations. The use of this technology has expanded the role of the traditional transportation practitioner into new areas of applying software, computers, electronic sensors, information technology, and communications to improve the efficiency of the transportation facilities. This is a significant change from the traditional capital development and the small signal systems projects of the past. A new set of skills and processes are required to harness these technologies (hardware and software) to the advantage of the agency. In addition to new technologies, Intelligent Transportation Systems are becoming interregional, with large numbers of stakeholders that need to work together. Individual agency systems now need to interface with other jurisdictions, forming larger regional systems. These institutional arrangements and cooperating systems require a higher degree of discipline to implement. A process is needed to successfully implement, document, and maintain these systems over a lifecycle of many years.

This Guidebook is intended to provide guidance in applying a disciplined approach to the development of ITS systems within an environment of rapidly changing technology.

2.4 Intended Audience

Table 2-1 below identifies the intended audience for this Guidebook. The intended audience includes agency project management team who will be responsible for the project from the time it receives agency approval until it is turned over to the operating organization. This team generally consists of a project manager, a lead project engineer, immediate staff and personnel from other organizations who will provide project support for example, procurement, finance, and contracts. It will be their job to manage and guide the activities of the team members of each task (either in-house or contractors) who will perform each of the systems engineering tasks.

The project management team will have the lead role in most of the systems engineering activities described in this Guidebook (the exceptions may be in the activities that take place after the system goes operational). Therefore, this Guidebook is aimed at providing that team with insight into these processes. This Guidebook will not only support them with a description of each of the steps of the systems engineering process but will help them understand and appreciate the goals of each step and the reasons that each step is important within the overall context of systems engineering. They will learn the flow of the processes, the inputs that must feed into each process and how the process outputs (products) are needed to support subsequent steps. The systems engineering activities found in this Guidebook are specifically focused on the successful development of Intelligent Transportation System projects.

Table 2-1 Intended Audience

Intended Audience Member	Benefit to be Gained from Guidebook
Project Manager	From this Guidebook, the successful ITS project manager will be able to identify the comprehensive, complete and necessary set of systems engineering tasks that have to be programmed into the project. The project manager's responsibilities include ensuring that all necessary tasks are part of the Project Plan, that the task deliverables are identified, that the resources needed for each task are identified and found, and that the requirements for all these tasks are reflected in the project's budget and schedule. An understanding of the lessons of this Guidebook will go a long way in supporting this responsibility.
Lead Project Systems Engineer	The Guidebook will support the lead project engineer in defining each systems engineering task so that each task not only provides the needed products, but that each task includes the specific systems engineering efforts needed to develop those products. For instance, given the specifics of a project, at various points there may be trade-off studies or engineering analysis needed to answer certain critical questions. The lead project engineer also may see the need to follow a unified set of systems engineering process techniques, both for efficiency and for quality of end product.
Project Technical Staff	The Guidebook will support staff in the best practices in systems engineering tasks where they have specific responsibilities. With this support, they will be better prepared to either perform their tasks, or, when required, oversee the performance of a task by the individual team members assigned to the task (either in-house or contractor). The Guidebook will support the staff in the development of each product of each task and recommendations of how that product must support the rest of the systems engineering activities.
Team Members Performing each Task (concept development, requirements, design, implementation, integration, verification, and installation.)	The Guidebook will support team members in understand the range of the other disciplines, the disciplines they will have to interact with. The Guidebook will support them in what is needed to and from these other disciplines. The Guidebook also will provide them guidance on the level and quality of their expected products, including documentation and technical reviews.
Project Team Management and other oversight / funding organizations	The Guidebook will provide an understanding of the systems engineering discipline that should be applied to a project to increase the chance of success. The Guidebook will help create in them both an expectation and a realization of the systems engineering tasks that should be a part of a well managed ITS project.
Planning Organization	This Guidebook will support Planning in the transition of the project from planning to project development. The Guidebook will provide support in verifying that the developed system is consistent with the regional ITS architecture and validation of the system against the concept of operations.
Operations and Maintenance	The Guidebook will support operations and maintenance in the planning for operations and maintenance. For planning and budgeting purposes, the Guidebook provides a checklist of key elements that will need to be addressed.
Owners of Interfacing Systems	The Guidebook will provide guidance for processes to be followed by the new system's project management and the role these system owners will be expected to play to insure technical and operational interoperability with their own systems.

2.5 *How to Use This Guidebook*

Figure 2-1 illustrates the organization of the Guidebook. The outer layer is the Executive Summary providing an overview of the Guidebook, and the next layer is a closer look at the systems engineering environment. Then the steps of processes and crosscutting activities are described followed by the foundation of roles and responsibilities, capabilities needed and finally example reference and supporting materials.

Understand the Guidebook and the systems engineering process (Sections 1 and 2). The first step is to understand the organization of the Guidebook, and the necessary steps of the systems engineering process. These chapters will point the reader to the relevant overview sections. Chapter 1 is the Executive Summary, which gives a short overview of the entire Guidebook. This is intended for managers or others who wish a quick view of the processes and key concepts presented here. Chapter 2 places the Guidebook into context in terms of purpose and scope.

Analyze and prepare the systems engineering environment (Section. 3). There are many factors that both support and constrain the systems engineering process for ITS. The Guidebook user needs to be familiar with these factors before starting work. Examples are the National ITS Architecture, FHWA Final Rule, ITS standards, and agency roles and responsibilities. This chapter also guides you in tailoring the systems engineering process to fit your particular project.

Form your team (Sections. 5 & 6). These chapters discuss the typical roles and capabilities of agencies, consultants, and developers. While such roles vary greatly from agency to agency, this will give guidance in putting together a project team.

Initiate crosscutting activities (Section. 4.8). There are several important activities that are ongoing continually or repeatedly throughout the systems engineering process. These include elicitation, project management, acquisition planning, generation of deliverables and documentation, process improvement, configuration management, interface management, risk management, program metrics, control gates, trade studies, technical reviews, and stakeholder involvement. These activities support the tasks carried out during the six phases.

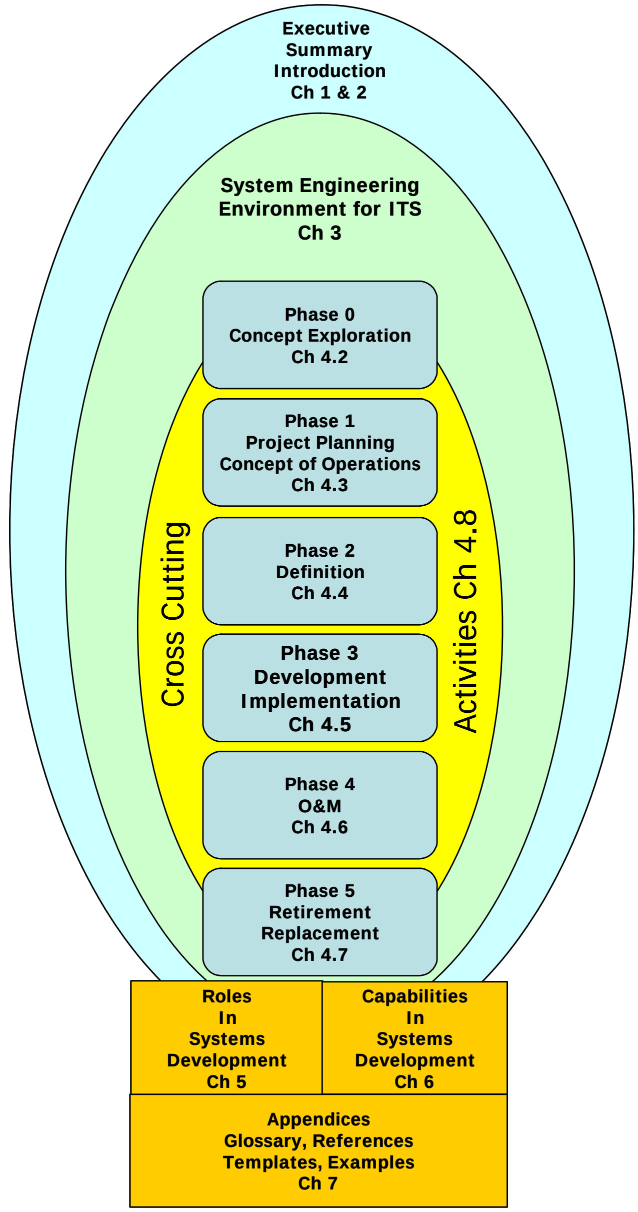


Figure 2-1 Organization of the Guidebook

Follow the systems engineering process (Section. 4.1 – 4.7). This is the heart of the Guidebook. The process follows six phases, as shown in the center of the diagram. Section 4.1 provides an overview, diagrammatic roadmap, and links to the key discussions in section 4. The other sections each correspond to the major phases of project development: concept exploration and feasibility assessment, project planning and concept of operations, system definition, system development and implementation, operations and maintenance, and retirement/replacement. A control gate that must be passed in order to proceed follows each phase.

2.6 Key Milestones and Project Time Table

The ITS Project Lifecycle on the following pages shows the entire set of systems engineering tasks required for an ITS project. All of these are described in detail in this Guidebook.

The entire sequence of systems engineering tasks is grouped into six phases (0 through 5) that cover everything from initial concept exploration to final system retirement. Within each phase are from one to four tasks. Each task is described according to its major activities, its primary products and its control gates. The section number in this Guidebook identifies each task.

Sequence of Phases, Tasks and Activities

Each of the ITS Project Lifecycle phases described in this Guidebook require a specific set of management and engineering skills. In large system development projects, activities within each task may be performed by a different set of people. For most ITS projects that this is not the case and the same individuals may perform several, or even all, of the activities within a task.

For these reasons, the phases and tasks in this Guidebook are to be performed sequentially, especially for phases 1 and 2. In these early phases, there is a temptation to start the next task prior to the completion and acceptance by the stakeholders of the current task. For ITS, this is not a recommended practice. It might appear that overlapping the tasks can shorten the schedule, but this introduces significant risks into the project. For example, starting the high level design prior to the development and acceptance of the system level requirements introduces the risk of reworking both, or worse the team moves on to detailed design prior to completing either of the previous phases. Within each task the activities are designed to work together to meet the objectives of that part of work. In some cases, similar activities can show up in different tasks or even phases. For example, prototyping is a primary activity at the detailed design task but prototyping also is often done in the requirement development task to ensure the feasibility of one or more requirements or as early as the Concept of Operations to validate concepts to the stakeholders.

Relative Durations of System Development Tasks

The following discussion only considers the duration of the system development activities, that is, the phases following Concept Exploration and culminating in Operations & Maintenance. System development refers to phases 1, 2, and 3 as illustrated in Figure 2-1. A detailed list of tasks, activities, products and control gates is located in section 4.1.1. Tasks performed before and after system development are subject to too great of variation (both for institutional and operations reasons) to make any generalizations on their duration meaningful.

Roughly speaking, phase 1 (Project Planning and Concept of Operations) takes about 10% of the total project duration, phase 2 (System Definition) about 35%, and phase 3 (System Development and Implementation) about the remaining 55%.

These relative levels of duration are useful as a rule-of-thumb or a reality check. The duration of every individual project must eventually be estimated on a bottom-up basis. That is, each individual task must be described in an appropriate level of detail. Then the time required for each task must be estimated based on the complexity of the individual task within the context of the specific project. Only then can a reasonably realistic schedule for a project be compiled.

3 Systems Engineering Environment

OBJECTIVE:

This section discusses a number of short sub-sections that affect the application systems engineering for Intelligent Transportation Projects. This section focuses primarily on institutional and project issues such as; how much systems engineering is needed to an ITS project and why, the relationship of systems engineering to existing agency Systems engineering practices, issues of procurement, and the relationship of Systems engineering to ITS standards, Transportation Planning and the ITS Architecture and Federal Final Rule.

The following sub-sections are discussed in this section:

3.1 Estimating the amount of systems engineering needed for ITS projects

This issue is addressed at the beginning of each project. There are a number of factors that need to be considered and the cost of the project is not necessarily a significant driver. The sub-section uses a scenario as an example of how much systems engineering is needed. But each project must be assessed on its own merit. Section 7.3.1, 2 and 3 provide details for the example.

3.2 Factors that drive the systems engineering environment

This sub-section describes factors that drive the need for systems engineering such as changing technology, maintaining your system, changing needs, stakeholder participation and flexible procurement.

3.3 Development models, strategies and systems engineering standards

This sub-section discusses various systems engineering models, strategies and their strengths and weaknesses and applicable standards.

3.4 Relationship of systems engineering to the national ITS architecture and the FHWA Final Rule

This sub-section discusses the relationship of this Guidebook to the ITS architecture and the FHWA Final Rule.

3.5 Relation to Transportation Planning

This sub-section discusses how traditional transportation planning relates to systems engineering and the bridge between Planning and ITS projects.

3.6 Relationship to ITS Standards

This sub-section discusses the relationship of systems engineering and ITS standards. It will look at the key ITS standards that systems engineering will use in the systems development.

3.7 Systems engineering support Environment

This sub-section discusses the importance of a good systems engineering support environment. This would include tools, processes, and training.

3.8 Common Agency Systems engineering activities

This sub-section discusses the existing systems engineering capabilities that may exist within the Agency and can be leveraged for ITS project development.

3.9 Systems engineering organizations

This sub-section discusses a typical systems engineering organization that can be used as a starting point. If an agency needs to establish one for their organization

3.10 Procurement Options

This sub-section discusses various procurement options that can be used for contracting systems engineering and systems development services.

In summary:

The following sub-sections amplify key issues that will be challenges to the application of systems engineering to ITS projects. These sub-sections are provided to give guidance and are not intended to be prescriptive. Each case will have exceptions and need to be reviewed and tailored on its own merit. These challenges will need to be factored into the agencies systems engineering support environment.

3.1 Estimating the Amount of Systems Engineering Needed for ITS

OBJECTIVE:

The objective is to provide guidance on how much systems engineering you will need for ITS projects. This section describes ways to assess the degree of systems engineering formality needed to deliver the system with the required level of quality.

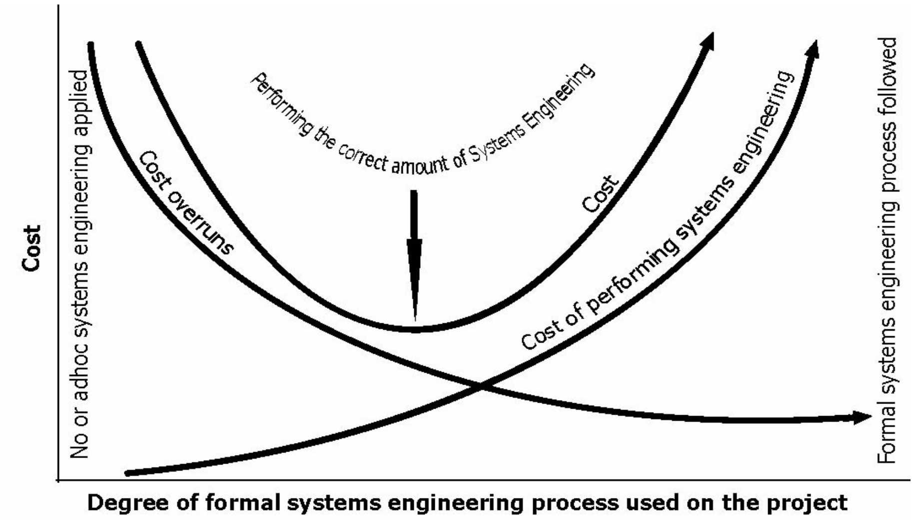


Figure 3-1 Degree of Formal Systems Engineering

How much systems engineering is needed for a given project? This is a difficult question to answer but one that needs to be addressed. Tailoring of the systems engineering process is a key initial activity for the systems engineer and it is not always based on a single factor like cost or size. In fact, there are a large number of factors that come into play, such as the number of stakeholders, the supporting relationships between them, complexity of systems (large number of interfaces to other systems, a level of functional integration, or the degree of coupling between systems.), level of ownership of system products (custom development of software owned by the agency or commercial off the shelf products), existing software products, resources, and risks. The amount of formal systems engineering is a question that cannot be fully answered quantitatively. Engineering judgment, experience, and institutional understanding are needed to "size up" a project. As the project is being carried out, the focus should be on the end product to be delivered and not on just a process to follow. In the ideal world the product delivered should carry just enough systems engineering process, documentation, and control to produce it at the desired quality level and no more or no less. There is a balance that needs to be kept and continually monitored. Illustrated in Figure 3-1 is a notional graph that depicts this balance. Mr. Ken Salter[1] adopted this analysis for systems engineering. It shows that one needs to look at each ITS project and assess the amount of systems engineering that needs to be done. This is the tailoring process that a systems engineering team performs as part of project planning.

The amount of systems engineering needed for a project depends on the following factors:

- Project risks
- System complexity
- Number of stakeholders
- Number of interfaces
- Decisions that need to be made
- Existing documentation

As a starting point for estimating the percent of effort, the Table 3-1 can be used.

Table 3-1 Estimate of Percent of Effort in SE

Planning	Definition	Design	Implementation	Integration/ Verification
10%	15%	20%	30%	25%

The following two examples are used to illustrate the amount of systems engineering needed in a very simplistic way. In section 7 there is an elaboration of each of these examples.

[1] LA Chapter INCOSE presentation 2003, JPL Pasadena

Example 1: Adding field elements to an existing system. (See Section 7.3.1 for more details) The following is a brief description of the example project:

> A $10 million project will add 30 full matrix Changeable Message Signs (assuming $330K per sign) to an existing system that has five of these same signs already deployed. No other changes to the central or field equipment are needed (including no required changes to the communications network). The system was initially designed to accommodate these additional signs so no additional software is needed. Assumptions are: 1) The communications and power for the signs have already been deployed under a previous construction project, 2) the initial system has been completed and the system is working, 3) the effort is limited to deploying the signs, installing the poles and foundations, procuring the controllers and wiring the controllers to the signs, 4) only configuration information about the signs needs to be added at the host by the user, and 5) the construction costs have been included in the cost of the signs.

In this example, even though this is a high dollar amount, a minimal amount of systems engineering is needed.

This example represents adding more of existing field equipment, for example, changeable message signs, cameras, traffic signals, and ramp metering, to an existing system. In this example the assumption is that the existing system was originally designed to accommodate the added elements and that no additional design work is needed. In this example the project risks are fairly low, there is a minimal number of decisions to be made, complexity of the project is low, common interfaces have been established, and minimal stakeholder issues exist. One may assume that the same documentation that implemented the original system is adequate for the expanded system. In this example little systems engineering is warranted other than a cursory review of system engineering products already delivered. However, it is prudent to review the existing documentation to ensure that no adverse effects will occur when the additional elements are added to the system.

Example 2: Builds on the first example but adds a new requirement for sharing control with another partner agency. (See Section 7.3.2 for more details) The following is a brief description of the example project:

> This new functionality was not pre-planned and assumes that new software will need to be developed and integrated into the existing system and the initial estimate for the software is approximately $500K to develop and integrate. Existing control software was not designed for this requirement and although the cost estimate is low with respect to example 1, it injected typical institutional issues that ITS projects face in developing regional systems. The point of this example was that the requirement for sharing adds significant risk to the project.

This example has introduced additional risks, additional decisions to be made, a broader set of stakeholders, and added complexity in functionality and interfaces. Further, there is the risk that the existing system cannot be easily changed to accommodate the new functionality. In this example the application of systems engineering is warranted to address these issues. It is interesting to observe that even though the cost estimate for this example adds only 5-6% more to example 1, the issues mentioned above and the addition of a custom development of software and changes to the existing system drove the need for much more formal systems engineering. The message is that cost alone is not a driver in defining the level of required systems engineering. Defining the appropriate level requires looking at a number of interrelated issues.

Since each project is unique, the recommendation is that each project must be assessed on its own merits as to the amount and degree of formal systems engineering that is needed. This tailoring is a systems engineering responsibility that occurs at the onset of each project.

Example areas of project tailoring

- Trade studies on the number of options to consider
- Number and formality of technical reviews
- Content of commercial off-the-shelf products in the system (the caution is that if the vendor is not qualified this may be a very high risk)
- Degree to which the system under development is similar to another. This will reduce the risks and reduce the elaboration of requirements needed for the second project.

Project tailoring is done as part of the project planning and is included in the Systems Engineering Management Plan (SEMP) which is described in Section 4.3.2.

3.2 Factors That Drive the Systems Engineering Environment

OBJECTIVE:

To describe the ITS factors that shape the systems engineering environment and describe how a systems engineering environment based on industry best practices can best serve the development, operations, and maintenance of Intelligent Transportation Systems.

Key factors that drive the systems engineering approach for Intelligent Transportation Systems (ITS) are 1) changing technology that impacts user needs, expectations, and project developments, 2) long term evolution and upgrades and 3) policy differences among partner agencies. As a result, the following are key challenges the systems engineering will need to address:

Rapid evolution in technology and tools

To keep pace with evolving technology and reduce the risk of overruns and schedule delays, it is recommended that technology choices be deferred until the last possible moment in the project development cycle and that short incremental development cycles be implemented. Complex projects should use an evolutionary development (evolve the system over time) which means that modular building blocks will need well documented interfaces.

Sustaining, maintaining and evolving the Intelligent Transportation System

The initial development is only the start of the lifecycle of an ITS. We expect these systems to be operated and maintained for decades and have the ability to evolve as the need changes. Systems engineering provides a disciplined way for the system to be documented and controlled. Systems engineering processes build-in system integrity during the development phase of the project and configuration management is used to maintain that integrity throughout the life of the system. The only way this can effectively happen is if the systems are well documented, that requirements are known and controlled using a change management process, that there is a high level of stakeholder involvement and buy-in, that design documentation is developed and accurately reflects the system elements, that standard interfaces are used, and that the system is well structured into modules.

Evolving needs of transportation

Systems engineering supports the evolving needs of transportation by maintaining a clear set of system requirements that are linked to the stated needs through the Concept of Operations. When needs change, this traceability will identify the areas of change and the impact to the system.

Participation of multiple agencies and a diverse set of stakeholders

The systems engineering process provides a clear roadmap for the development of systems; when adapted the stakeholders are aware of the steps and can understand what is expected during any phase of the project. Participation of stakeholders is facilitated when everyone is on the same page of the project and has a common language and understanding.

Development of regional and state ITS architectures

The development of a regional and state ITS architecture is a starting point for the development of ITS projects. (Architecture here means the framework that was set up for the region and not a project architecture that can be built). The regional and state ITS architectures provide the initial set of stakeholders, needs, inventory, operational concepts and requirements that define the roles of the various agencies. These elements flow directly into the systems engineering process for the project level Concept of Operations and requirements. These high level inputs from the architecture are then refined into project level requirements that the developer can implement.

Flexibility in procurement options for consultants and development teams without sacrificing system integrity

Systems engineering provides the system owner the greatest flexibility in contracting options. When the systems engineering process is implemented and used, the products from the project are well documented. When the system needs to evolve, change, or be upgraded, the system owner has the option to select from a number of qualified consultants and development teams and is not locked into a particular consultant or contractor. It is recommended that the system owner choose consultants that have systems engineering experience and development teams that have documented internal processes and both can demonstrate performance in applying systems engineering. (See Section 6 for additional information)

3.3 Development Models, Strategies and Systems Engineering Standards

OBJECTIVE:

To describe the key industry-standard systems development models, to describe the purpose of each, and why it is important to use them. This section will also talk about project development strategies which are methods for evolving the system over time.

Models for systems development are important for the following reasons:

- Illustrates a common framework for the team and stakeholders
- Describes relationships between activities

Models for the systems and software development have been depicted in two principal forms: the Waterfall development model first described in 1969 (Win Royce) for systems with software components (see Figure 3-2) and the Spiral model described in 1983 (Barry Boehm) for risk reduction in software developments (see Figure 3-3). These two models are still the foundation for systems and software development.

In 1988, National Aeronautics and Space Administration (NASA) saw a benefit of bending the Waterfall model into the "V" shape for software development. This was the original Vee technical model as shown in Figure 3-4. In 1990, Kevin Forsberg and Hal Mooz created an enhanced version of the Vee model integrating the best aspects of the Waterfall and the Spiral development models. By adding emphasis on risk, opportunity, and stakeholder involvement and by augmenting the Vee with a development strategy (iterative/evolutionary development features), the goal was to develop a general case development model for systems. This was a departure from the previous models, which had a focus on software. The Forsberg and Mooz Vee model was published slightly after the NASA model in October 1991 at the NCOSE (National Council on Systems Engineering) symposium in Tennessee. Since then, the Vee Development Model has become widely accepted and is illustrated in some form in both the EIA 632 and ISO 15288 systems engineering process standards. In addition, this model is being adopted in the broad spectrum of systems development environments.

The following are some observations of the Waterfall, Spiral and Vee development models.

3.3.1 The Basic Waterfall Development Model

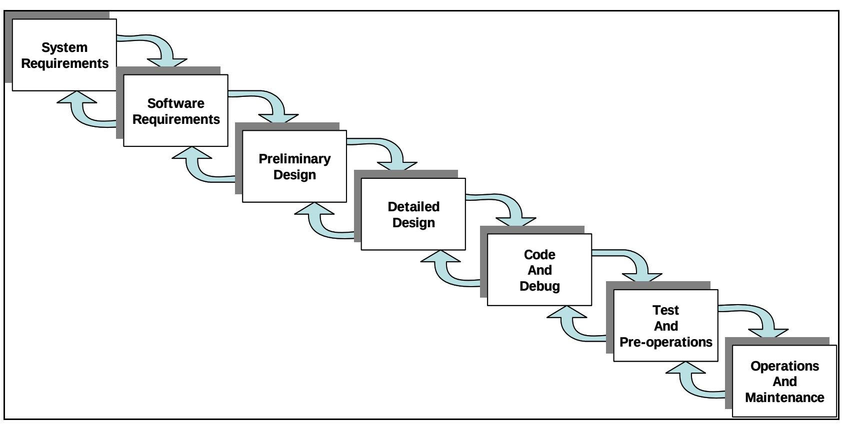


Figure 3-2 Waterfall Development Model (Royce 1969)

Brief commentary on the waterfall development model:

- Initial development model for software systems development
- All requirements are known up-front
- Form follows function philosophy – What to do (function) before How to do it (Form)
- Still used for certain types of systems – systems with low complexity and systems that cannot evolve.
- Relationships between the early phases of the project to the end results are not illustrated
- Stakeholder involvement is not recognized beyond the initial requirements
- Control Gates not always clear

3.3.2 Spiral Development Model

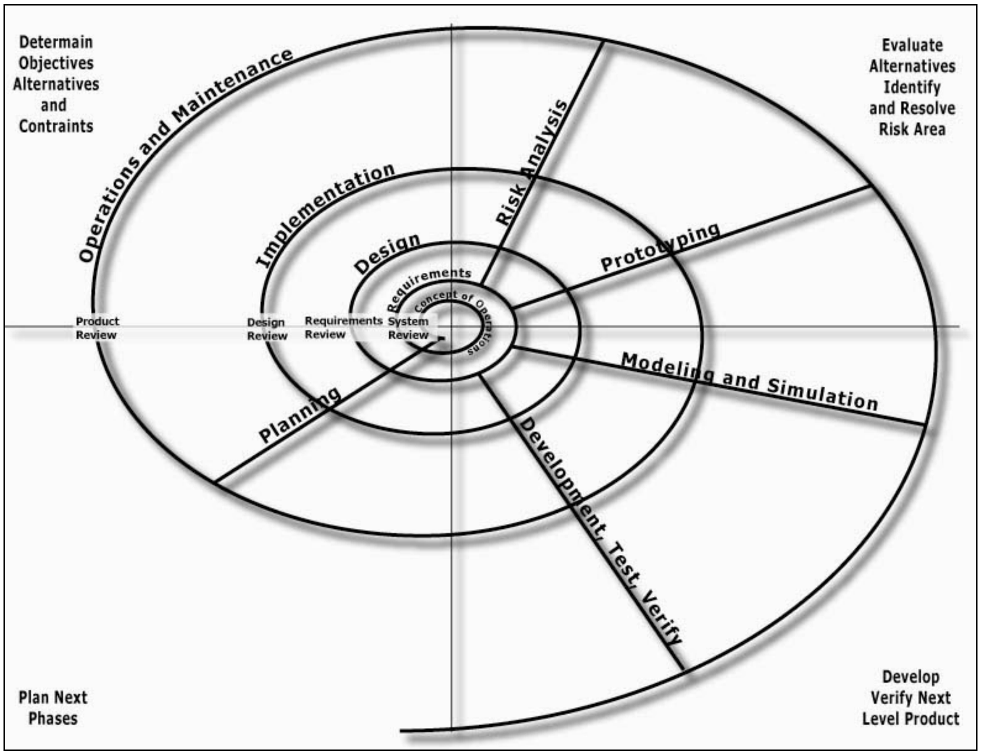


Figure 3-3 Spiral Development Model (Boehm 1983)

Brief commentary on the spiral development model:

- The goal of the model is mitigation of software development risk
- Emphasizes the need to iterate between form and function experimentally
- Popular in software development – It deals nicely with emerging properties and partial solutions of software such as user interfaces, algorithms, or alternative sequences of events. This is sometimes called the "I Know It When I See It" (IKIWISI) approach.
- The spiral principle leads to the evolutionary approach to systems development illustrated in the Vee development strategies.
- This model can be used within the phases of the Vee Development Model to examine the feasibility of a concept or to derive or clarify a set of requirements
- Does not include the concept of decision/control gates or the baseline of documentation. This approach does not promote the idea of developing a complete set of documentation and it is easy to lose the synchronization between the documentation with the actual software product.
- Minimizes the idea of defining the goals up front, and encourages never-ending cycles of development. "When have you finished the product?"

3.3.3 Vee Development Model

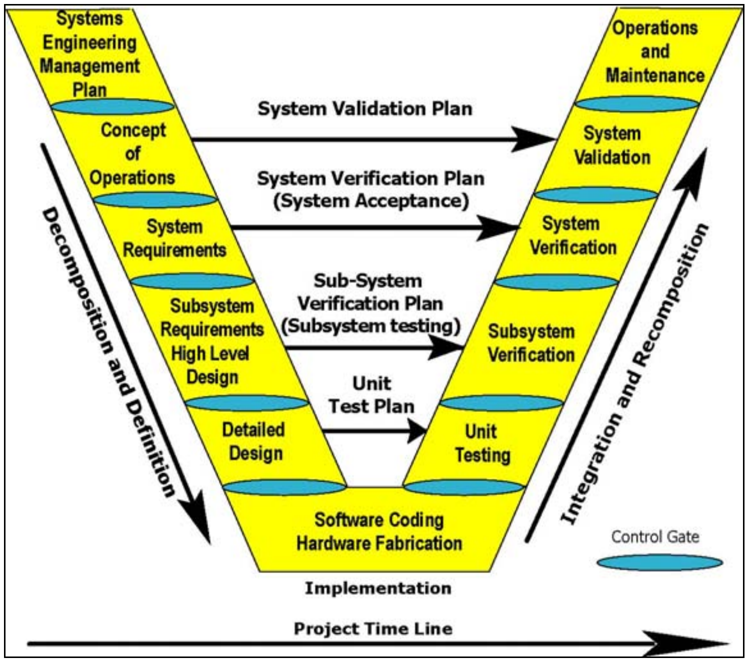


Figure 3-4 Vee Development Model (Kevin Forsberg, Hal Mooz 1991)

Brief commentary on the Vee Development Model:

- Illustrates the influence of the early phases of the project with the end of the project
- Emphasizes the planning, and stakeholder involvement and validation of the requirements as well as the validation of the product
- Illustrates the relationship of the model of the system to be built (left side) with the realization of the end product (right side)
- Illustrates the planning, defining, and performing integration and verification. Emphasizes the need to begin verification planning at the time requirements are first defined at every level.
- Encourages the “Starting at the Finish Line” mindset, by looking at the validation of the product at the same time as developing the Concept of Operations as well as the development of Verification Plans with the requirements at every level.
- Encourages definition and control of the evolving baseline at each phase of the project
- Illustrates “top down” definition and decomposition (the breaking down the project architecture into small building blocks from the top most level to the lowest component then a specification is written for it to be built is a key systems engineering activity) and “bottoms up” building, integration and verification (building the developed components up in a step wise manner from the components to the top most system)

A complete description of the Vee technical models is provided in section 4.1.1.

3.3.4 Development strategies

The following development strategies are different ways that a project is implemented and completed. They include:

- ***Single evolution*** Figure 3-5**.** Single delivery; one single pass through the Vee
- ***Incremental with single or multiple deliveries*** Figure 3-6. Breaking a project up into sub-systems as independent developments. Single delivery integrates the sub-systems and delivers a completed system. Multiple delivered subsystems are integrated into an operational environment. (See below for examples of each)
- ***Evolutionary development*** Figure 3-7**.** Breaking a project down into parts and developing them in serial fashion e.g. phase 1 to develop and deploy the servers, software and communications, phase 2 to develop and deploy the workstations and software, and phase 3 to develop and deploy the field devices and software.

One can mix and match these tactics into a hybrid approach, such as having an evolutionary development in which each evolution can be incremental with single or multiple deliveries.

The strategy that is selected for development is usually driven by one the following conditions:

- Funding level – project built in multiple phases to accommodate funding increments
- System size and complexity – large projects broken down into manageable developments
- Institutional issues – inter-agency agreement on interfaces, operations, and maintenance responsibilities and consensus on system features.

The selection and tailoring of the strategy is done during the project planning phase or before. If funding is the driving factor, the agency may choose evolutionary development because of yearly funding increments. With large, complex projects, and the need to get the project deployed quickly, agencies may elect to use the incremental strategy where sub-systems are developed by different development teams and then brought together.

The following are observations on each of the strategies mentioned:

Single Evolution - Single Delivery

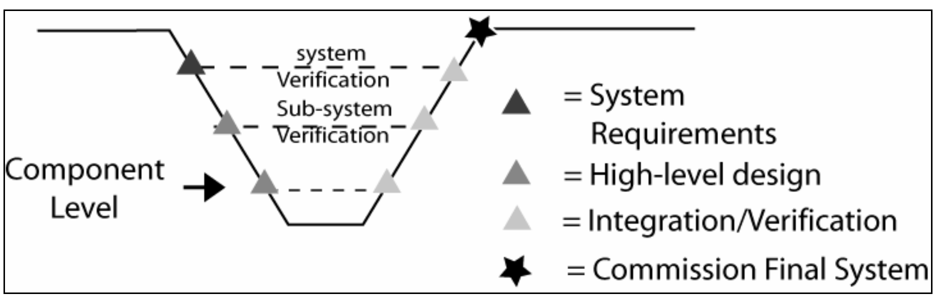


Figure 3-5 Single Evolution – Single Delivery

Brief commentary on single evolution- single delivery

- All requirements must be known up front
- Used on simple projects having only a few requirements
- Used on projects that cannot evolve or that need to be developed in a single pass
- This was the classic development strategy in the early days of large military projects
- No longer recommended in developments that can evolve

Example ITS projects that may consider this strategy:

- Signal control system
- CMS, CCTV, detection sub-systems
- Small incident management systems
- Single agency minor ITS projects

Incremental Development: (single delivery and multiple)

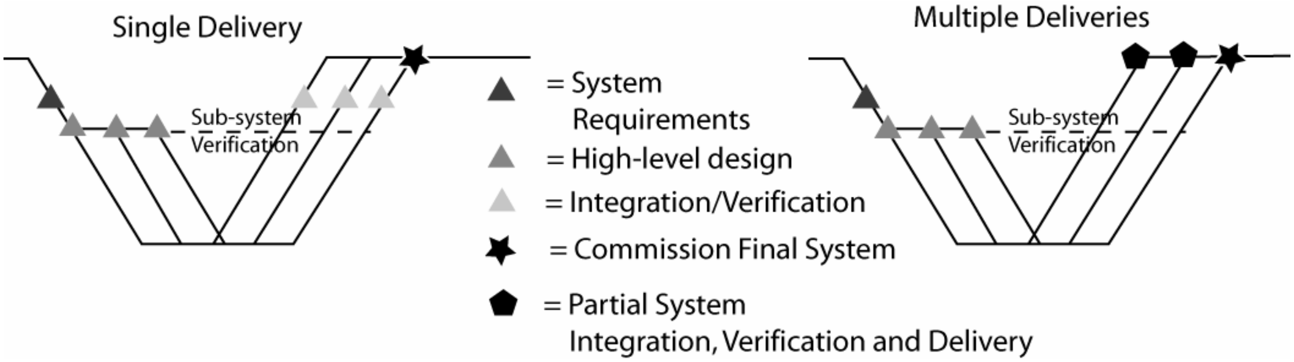


Figure 3-6 Incremental Development with single and multiple deliveries

Brief commentary on incremental development with single or multiple deliveries

- Used on large systems that can be divided into clear sub-systems
- Works with multiple development teams
- Used when significant or full system capabilities can be delivered by the sub-systems one at a time and offer useable capabilities on their own.
- Need a significant amount of coordination between projects to ensure integration
- Risk of finger pointing if different development teams are developing different part of the system
- Use of multiple deliveries would be if each increment can be completely verified in a stand alone configuration.
- Use of single delivery would occur if there are dependencies between the increments that need be verified prior to delivery.

Example ITS projects for single delivery strategy:

- o Reversible Control lane system
- o Communications infrastructure (major subsystem)
- o Toll Collections system (Major subsystem, collection system, tag processing, enforcement)

This strategy is used for systems or major subsystems that need to be fully functional before being deployed into service.

Example ITS projects multiple delivery strategy:

- Traffic signal system
    - o Central management system followed by:
    - o Intersection group 1 (1-5) then
    - o Intersection group 2 (6-10) then
    - o Intersection group 3 (11-15)
- Motorist information systems
    - o Central management system followed by:
    - o Distribution to partner agencies then
    - o Internet service providers etc
    - o Extending additional changeable messages signs to an existing control system

This strategy is used when partial expansion of existing system can be deployed over time. It should be noted that in the case of the multiple delivery strategy, the initial subsystem - in this example, the Central management system for both the Traffic Control and Motorist information system - needed to be fully functional using the single delivery strategy and the expansion of the system elements followed using a multiple delivery strategy.

.

Evolutionary Development

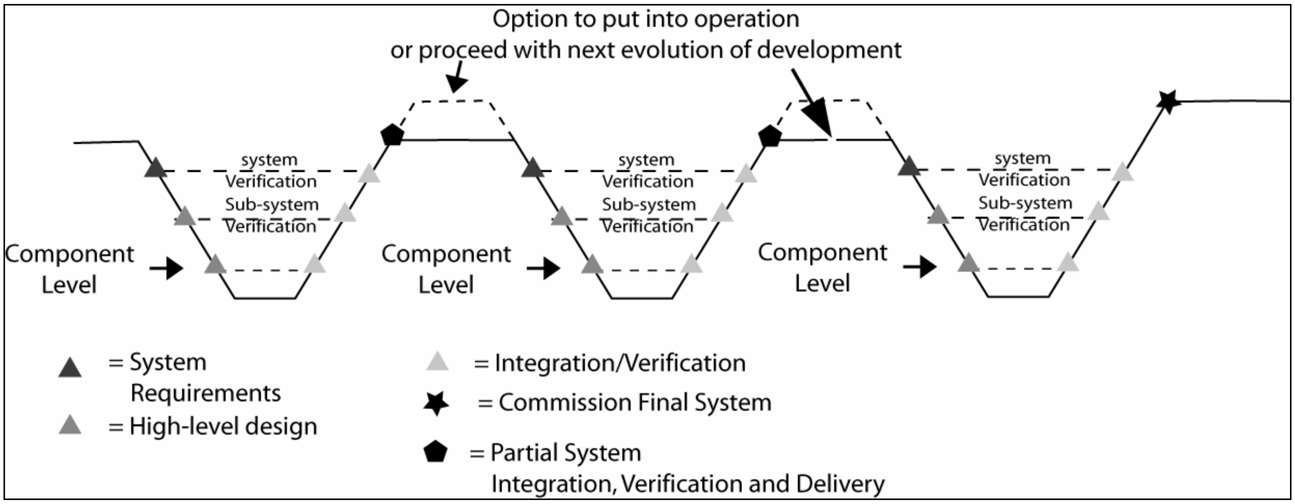


Figure 3-7 Evolutionary development

Brief commentary on evolutionary development

- Used when funding is limited but can be obtained in multi-year cycles
- Large projects where not all of the requirements are known but enough are known to build initial capabilities
- Multi-regional systems in which stakeholders are going to need to develop systems internally to join with a broader set of stakeholders
- Where institutional issues are complex and some initial capabilities are needed to resolve them
- Projects may or may not provide capability that will go into service but will be building blocks for the next evolution of the system
- Currently the strategy of choice and recommended when possible

Example ITS projects that may consider this strategy:

o <u>Incident Management System (single or multi-agency)</u>

Possible Sequencing:

Subsystem 1-The Communications backbone

Subsystem 2-Surveillance (CCTV)

Subsystem 3-Changeable Message signs

Subsystem 4-Detection system

Subsystem 5-Incident Management Software

o <u>Regional Advanced Motorist Information System</u>

Possible sequencing:

Subsystem 1-The Communications backbone (agency interfaces and agreements)

Subsystem 2- Detection system

Subsystem 3- Data Process software

Subsystem 4- Media Interfaces

Subsystem 5 Surveillance (CCTV)

Subsystem 6-Video interface to Media

Note:

Any of the projects that were done incrementally could be done in an evolutionary deployment way. In some cases, it may take several evolutions of development before it is ready to be commissioned into service. The interim evolutions would not be put into service until the whole system has been completed. For example a reversible lane control system may be implemented using the evolutionary deployment but would not be commissioned into service until all of essential sub-systems have been developed and integrated.

3.4 Relationship to the National ITS Architecture and FHWA Final Rule

OBJECTIVE:

Describe the relationship of the National ITS Architecture and the FHWA Final Rule to the ITS systems development process described in this Guidebook.

National ITS Architecture

The Federal Highway Administration (FHWA) requires ITS projects using Federal funds from the Highway Trust Fund, including the Mass Transit Account, to conform to the National ITS Architecture via a regional ITS architecture. The National ITS Architecture provides guidance for the development of ITS projects. It provides a flexible template of interconnections and interfaces to select from at the regional level. In fact, it provides a full range of elements that may be used as ideas or starting points for the Concept of Operations and requirements.

The National ITS Architecture is derived from ITS User Services. These provide a catalog of features that could be provided by ITS projects for public or private users. Each has associated baseline requirements. They are organized into 8 bundles:

Travel and Traffic Management	Emergency Management
Public Transportation Management	Advanced Vehicle Safety Systems
Electronic Payment	Information Management
Commercial Vehicle Operations	Maintenance and Construction Operations.

User Service Bundles

The Market Packages address specific services like surface street control. They suggest ideas for subsystems to provide selected services. They are organized into 8 Service Areas:

Archived Data Management	Vehicle Safety
Public Transportation	Commercial Vehicle Operations
Traveler Information	Emergency Management
Traffic Management	Maintenance and Construction Management

Market Packages

A complete description of the National ITS Architecture is available from the USDOT ITS web site at http://www.its.dot.gov/arch/arch.htm.

The FHWA Final Rule on Architecture Standards and Conformity (Final Rule) requires the development of regional ITS architectures and that all ITS projects using Federal funds be developed using a systems engineering analysis.

The elements of the Final Rule are as follows:

- 940.5: Describes the requirement to use the National ITS Architecture to develop regional ITS architectures, and the need for consistency with transportation planning processes;
- 940.7: Describes the specific applicability of the regulation;
- 940.9: Describes the specific requirements for developing and regional ITS architecture;
- 940.11: Describes the specific requirements for a systems engineering analysis;
- 940.13: Describes the project implementation requirements; and,
- 940.15: Describes the requirements for project oversight.

Section 23 CFR 940.11 specifies certain activities that are to be performed to accomplish a systems engineering analysis. They follow with notation where this Guidebook will help with each.

1. Identification of portions of the regional ITS architecture being implemented (or if a regional ITS architecture does not exist, the applicable portions of the National ITS Architecture) [see 4.2.1 in this Guidebook];
2. Identification of participating agencies' roles and responsibilities [see 4.3.3 (Con Ops) & Chapters 5 and 6];
3. Requirements definitions [see 4.4.1];
4. Analysis of alternative system configurations and technology options to meet requirements [4.2.3, 4.4.2, and 4.4.3];
5. Procurement options [4.8.7];
6. Identification of applicable ITS standards and testing procedures [3.6]; and
7. Procedures and resources necessary for operations and management of the system [4.6.2].

State and Local Agency Programs

State DOT's lay out the way for transportation agencies to show evidence of meeting the FHWA

Final Rule. These procedures will vary from state to state. Most states have offices that specifically manage federal funding for local agencies and establish procedures for receiving funding.

There are often other state and regional regulations that guide project development. They are too numerous to discuss here. Be sure to familiarize yourself with the applicable regulations before starting the project. The project will need to be compliant with them.

While this Guidebook has attempted to present a process that is applicable everywhere, there is no guarantee against conflicts between this book and local regulations. In all cases, the regulations take precedence.

3.5 Relationship to Transportation Planning

OBJECTIVE:

Describe the role of transportation planning to the project level systems engineering process. Identify the relationship and role that planning has in ITS developments.

For State and local transportation agencies and metropolitan planning organizations, comprehensive planning is a critical element in the development of Intelligent Transportation Systems. Planning professionals take a leadership role in developing the regional ITS architecture, which sets the framework for future projects and sets the stage for individual projects to be developed and integrated together. The regional ITS architecture is intended to look at the big picture for the region and how individual projects will work together. The output of this strategic planning activity provides the foundational input to the project level development. In addition to the traditional early planning activities, development of regional and state ITS architecture is strategically performed before project identification, and programming into the Transportation Improvement Plans (TIP) for funding. The roles will be covered in Section 4.2.

Participation by planning professionals in the early stages of system project development is also important. Their perspective on resources, budget and timeline, helps strengthen the Concept of Operations documentation of the way the system will be used from multiple viewpoints. These roles will be covered in Section 4.3.

The following is a comparison of roles played by the traditional DOT divisions in capital ITS infrastructure projects as compared to their roles in ITS system developments.

The role that the planning department currently plays in the development of capital ITS infrastructure projects is analogous to the use of the left side of the Vee Development Model for ITS system developments (See section 3.3) – only it's performed at a higher (regional or program) level. Stakeholder needs are identified, the system and problem space is modeled, alternatives are explored, and requirements for the project are defined. When the projects are defined by planning they are then placed into the TIP. Upon completion of this strategic process a transitional hand-off to Project Development occurs. Planning then becomes minimally involved in the design and implementation of the individual projects. In concert with Traffic Operations, Project Development design and implement the project, Traffic Operations manage the project, and the Maintenance division maintains the facility and supports traffic operations. These roles are well defined.

For ITS system development projects a different pattern surfaces. The Planning division provides their traditional role in early project planning, including the development of the regional ITS architecture. From this point there is often an activity undertaken, usually by the Traffic Operations division, to perform a feasibility analysis. Traffic Operations then addresses the more specific process steps that make up the left side of the Vee Development Model. These include identifying the more specific needs of the system user and breaking down the definition of system and subsystem requirements. As was stated earlier, these definition steps before actual design are analogous to traditional Planning strategic steps, except at a more specific project development stage. This should not exclude Planning's participation. Although the traditional handoff has occurred, planning stays involved through at least the user needs stage known as Concept of Operations. This will be further discussed in Section 4.3.

Information Technology departments have also become more involved with the implementation, deployment, and maintenance of these systems. They have largely been introduced to the project with the development of any feasibility study report that may be required as discussed in Section 4.3. Additionally the Maintenance division, who supports operations and maintains primarily the field elements, should be involved in the early stages of definition.

In summary, for Intelligent Transportation Systems developments, it is important that an integrated view be adopted for the development, operations, and maintenance of these systems. This integration must have a clear and inclusive interface between Planning and ITS system development. Table 3-2 illustrates the point of interface that exists between Planning / Regional ITS Architecture and Systems Development / Project Development, and the Bridge between them.

Table 3-2 Bridging Between Planning and Systems Development at the Project Level

Planning	Systems Development	Comments
Regional ITS Architecture	***Project Development***	***Bridge between Planning and Development***
Inventory	Concept Exploration and Feasibility Assessment Concept of Operations	Existing systems and legacy interfaces
Stakeholder Identification	Concept Exploration and Feasibility Assessment Concept of Operations	Starting point, additional project stakeholders need to be added, e.g. maintenance, operator, and managers.
High Level Needs/Services	Concept Exploration and Feasibility Assessment Concept of Operations	Goals and objectives for the regions; specific project level goals must support these.
Area of Coverage	Concept of Operations	Forms the boundary for the projects of the architecture
Operational Concept	Concept of Operations	Identifies the initial roles of the stakeholders
High Level Requirements	Concept Exploration and Feasibility Assessment Concept of Operations System Level Requirements	Starting point for requirements. These requirements will need to be refined for each of the projects making up the regional ITS architecture
Interconnect/Information Flows	Concept of Operations Requirements and High Level Design	Provides the initial set of interfaces for the projects. These will need to be refined at the project level based on the tailoring of the service
ITS Standards	Requirements and High Level design	Identifies a set of candidate ITS standards that can be used for interfaces
Project Sequencing	Project Planning Concept of Operations	Defines the evolutionary path
Interagency Agreements	Concept of Operations High Level Design	Defines stakeholders' role in operations and maintenance, Interface Control Documents

3.6 Relationship to ITS Standards

OBJECTIVE:

Identify the relationship between this Guidebook and ITS standards. Since the focus of this Guidebook is on System Engineering, this section will identify key systems engineering process standards and other related ITS standards. This section will also briefly talk about ITS protocol and equipment standards that are evolving.

Why use ITS Standards?

Don't reinvent the wheel – Use of an equipment standard (for a Dynamic Message Sign, for instance) means that you don't have to specify the requirements from scratch. However, be aware that equipment standards may not keep up with advances in technology.

Avoid early obsolescence – By gradually migrating field devices to ITS standards compliant devices you are moving your system in the direction the industry is going.

Obtain a choice of vendors – Products conforming to an ITS standard can be interchangeable with products from other vendors. However, interchangeability is hindered by vendor specific features that go beyond the standard or by partial implementation of a standard.

Multipoint control of devices is going in the direction of IP-based networks. Not only will this put all devices on a single network, but it also means that any center on the network can access and control any device. This allows the centers to back each other up in case of failure or operational downtime.

Potential Benefits of Standards to Systems Engineering Processes

If you are building a system that has components covered by mature ITS standards and if the existing ITS standard supports your operational concept, then use of ITS standards can be of considerable benefit to many, if not most, of the systems engineering processes described in this Guidebook. Obvious examples include:

High Level Design and Component Level Detailed Design – If an ITS standard (say for a Dynamic Message Sign) can support your requirements, then use of such a standard eases your design tasks and allows you to use predefined proven components

Hardware/Software Development – Use of ITS standard components, like use of any off-the-shelf product, will reduce the design effort.

Integration and Verification – If the chosen ITS standards are mature, then both integration and verification efforts will be easier. On the other hand, if the ITS standard is not mature, or has not been used before, the effort to prove the product matches the standard can be trying.

Risk Management – Use of a proven product built to a mature ITS standard will reduce development risk in a project.

Procurement Options – Use of ITS standards will make it easier to specify the product you want and allow multiple vendors to compete to provide the same standard product.

Standards are widespread in the transportation industry and are generally developed for one of two reasons: to improve interoperability or to stimulate competition. For ITS, a primary emphasis in the standards being developed is on the interoperability of systems and the interchangeability of subsystems and components. This leads to easier system integration and smoother coordination among systems.

What does FHWA Final Rule (23 CFR 940.11) say about the use of ITS Standards?

FHWA Final Rule (23 CFR 940.11) requires that "All ITS projects funded with highway trust funds shall use applicable ITS standards and interoperability tests that have been officially adopted through rulemaking by the DOT." As of the date of this writing, while the DOT recommends judicious use of the available standards, none of them have been officially adopted through rulemaking. The FHWA Final Rule also expects the regional ITS architecture to identify "ITS standards supporting regional and national interoperability" and the Final Rule expects consistency between the regional ITS architecture and any related projects.

NTCIP Standards Development

One ITS standards effort is by an organization called National Transportation Communication for ITS Protocol, or NTCIP. These standards identify protocols and message sets to be used Center to Centers, Center to Roadside, and Vehicle to Roadside. It is a joint standardization project of the American Association of State Highway and Transportation Officials (AASHTO), the Institute of Transportation Engineers (ITE), and the National Electrical Manufacturers Association (NEMA), with funding from the U.S. Department of Transportation (USDOT).

(See NTCIP 9001 National Transportation Communications for ITS Protocol – NTCIP Guide Version 3 at Website http://www.ntcip.org)

Other ITS Standards Activity

Standards Development Organizations at the national level that are working on ITS standards also include:

American National Standards Institute (ANSI), http://www.ansi.org (general communications)

American Society for Testing & Materials (ASTM), http://www.astm.org (Vehicle to Roadside)

Institute of Electrical and Electronics Engineers (IEEE), http://www.standards.ieee.org (Center to Center transit and incident management)

Society of Automotive Engineers (SAE), http://www.sae.org/topics/itsinits.htm (Center to Center and Vehicle to Roadside)

Documentation and Process Standards Activity

Another area of standard development that is of use to the systems engineer involves documentation standards and systems engineering process standards. Systems engineering document and process standards offer the systems engineer good advice in the following areas:

Systems Engineering Process

> Institute of Electrical and Electronics Engineers, IEEE Std. 1220-1998 IEEE Standard for Application and Management of the Systems Engineering Process

> Electronics Industries Alliance, EIA 632 Standard Processes for Engineering a System

Concept of Operations Document see section 4.3.3)

> IEEE 1362 IEEE Guide for Information Technology – System Definition – Concept of Operations Document

Concept of Operations Document (see section 4.3.3)

> American National Standards Institute / American Institute of Aeronautics and Astronautics, ANSI / AIAA G-043-1992 Guide for the Preparation of Operational Concepts Documents

Requirements Specifications (see section 4.4.1)

> IEEE STD 1233 IEEE Guide for Developing System Requirements Specifications

Configuration Management (see section 4.8.6)

> EIA 649 National Consensus Standard for Configuration Management

Technical Reviews and Audits (see section 4.8.12)

> IEEE 1028-1988 Standard for Software Reviews and Audits

Software Architecture Design (see section 4.4.2 and 4.4.3)

> IEEE 1471-2000 Recommended Practice for Architectural Description of Software-Intensive Systems

Independent Verification and Validation (see section 4.5.3 and 4.6.1)

> IEEE 1012-1998 Standard for Software Verification and Validation

System Modeling Standards Activity

Over the years there have been many attempts to develop modeling approaches to help with system and software design, including:

Unified Modeling Language

This is a language for specifying, visualizing, constructing and documenting the design of software. The standard for UML is maintained by the Object Management Group and information on UML is available at their web site, http://www.omg.org

Integrated Method for Information Modeling (IDEF)

Another method for modeling processes is called IDEF. This technique is used in the Guidebook to model the processes described in Part 4. Information on IDEF can be found at http://www.idef.com

3.7 Systems Engineering Support Environment

OBJECTIVE:

The systems engineering environment needs to support the systems engineering capabilities within the agency. This section describes basic support needs for the systems engineering environment such as the development of a documented process, process improvement, training and capacity building, technology reuse, and systems engineering support tools to carry out the documented process.

One of the keys to success for ITS projects will be management support and an environment that promotes the use of the systems engineering process for the development of ITS projects. A well-defined and documented process, tools, training, and the application of technology across agency projects will be important to the success of projects. The following elements describe this environment.

The systems engineering environment needed to support successful project development includes the following key elements:

Defined and documented process and process improvement

Documented systems engineering processes must support the organization's internal goals and objectives. It is recommended that a documented set of systems engineering processes be created and this Guidebook would be a good starting point for those procedures. The use of a common set of processes will benefit ITS, as the established set of processes has for capital projects and in the same way. Once the systems engineering processes have been developed they will provide a common framework by which ITS projects are carried out, one that will benefit the agency in the utilization of their resources and their ability to pull together teams efficiently for projects. In addition to these processes, a method is needed to assess how well the process is accomplishing its intended purpose and then to adjust the process continuously to improve its effectiveness. (See process improvement).

Capacity building and training development

Training will benefit an agency in development of capabilities in key systems engineering topics and should be part of the systems engineering environment. This training includes both in-house and contracted training courses. Training in contracting, project management, systems engineering, configuration management, risk management, and maintaining the regional architecture are some of the basic courses that are recommended for ITS practitioners. Other specialized courses, such as requirements engineering, reverse engineering, modeling and simulation, architecting, and software and hardware design, should be considered for staff that will be focusing in these areas. Since technology is changing, refresher classes in all of these areas are recommended.

Technology transfer

Organizations can benefit and optimize the use of technology by being aware of the technologies that are in use throughout the organization. Organizations must assess vendors to ensure their ability to produce quality products that will be supportable, maintainable, and affordable for the projects. Standardization is a way to reuse technology and minimize the risks of new developments.

Systems engineering support

Systems engineering support provides the environment to enable various aspects of systems engineering to be performed. These tools may include, for example, requirements management and modeling tools, test beds, simulators, training, office space, documented processes, software, and test equipment.

Process improvement

An organization should provide for the continuous process improvement to fine tune the processes over time. Initially an organization will put into place a set of processes and procedures and use a test case project to wring out the steps in the process. Then it will re-write or modify the weak areas found in the processes or where a process is found to be too rigid and costly. The process may be relaxed to fit the real world situation. Over time, this process becomes part of the support environment and is continuously improved with the lessons learned on each project.

3.8 Common Agency Systems Engineering Activities

OBJECTIVE:

Synchronization between the processes in this guidebook with other agency systems engineering activities is described in this section. These common activities exist in some form within most agencies. This Guidebook is intended to leverage these processes, to complement and not duplicate or be in conflict with these processes. The following is a description of some common agency activities that can be leveraged for the systems engineering processes.

Configuration Management (CM) activities

Agency level configuration management is a function that is responsible for monitoring and approving changes to the hardware in the field, for example signal controllers, and communications. In some agencies this may come from the Information Technology department, or it may be called asset or resource management. This could be leveraged to perform configuration management for Intelligent Transportations Systems (ITS). Their processes and procedures would need to be augmented to manage ITS systems development and operations and maintenance. If these procedures are not in place, a configuration management capability at the agency level will need to be developed.

Standardization

Applicable agency standards from the Information Technology department should be leveraged for the systems engineering process. These standards may constrain the developers on technology choices such as databases, software applications, workstations, and servers. This may be of great benefit when purchasing software licenses, or workstations, or choosing a databases and operating systems. It also may have a disadvantage in that ITS applications will be constrained to these choices and preclude better or more efficient solutions for the designers.

Feasibility process

Agencies often evaluate alternative solutions to choose the best cost/benefit solution and justify the business case for a project. These activities may have a strict internal processes defined. If available they will be used for their ITS projects during the early planning stages. For example, the State of California uses the Feasibility Study Report for the justification of IT and ITS projects. The products from this process may be used for the system engineering process for the project such as the goals and objectives, vision, stakeholder lists, and key performance measures. Again these processes may need to be tailored to satisfy agency policy requirements.

Information Technology process and guidance activity

The Information Technology (IT) department of an agency may have resources that can be leveraged for Intelligent Transportation Systems developments. Most IT departments have development processes in place that focus on similar information technology applications. These same processes may be adapted for ITS. Since systems engineering integrates different disciplines, the leveraging of Information Technology processes needs to be evaluated using other domain expertise such as traffic operations. It is critical that domain expertise is involved with the tailoring of these processes.

Systems engineering capabilities for small and large agency organizations

For a small local governmental agency implementing a single ITS project, systems engineering exposure may be minimal; (See 3.1) it may be adequate to have the system owner take some systems engineering training in systems engineering fundamentals and then tailor, implement, and manage the SE processes by themselves. Another option is to hire a consultant experienced in systems engineering (See 3.10) to perform these activities, or get available assistance from the State DOT and/or the FHWA resource centers. This support environment may be temporary and only needed for a specific project.

Larger organizations, for example an MPO or State Transportation Agency, will benefit from an established systems engineering support environment and leveraging from the existing agency activities across all of the projects. This “umbrella” systems engineering experience within an agency can lead to a number of valuable services for these projects, Some examples are the sharing of appropriate skills needed to carry out the roles and responsibilities of each project, generally sharing experiences through lessons learned, independent technical reviews, an established common approach, sharing technology and tools, and re-use of project products.

3.9 Systems Engineering Organization

OBJECTIVE:

Describe typical systems engineering organizations and the role that these organizations play in the development of ITS

What Makes an Effective Organization?

Effective systems engineering requires an effective organizational structure. That structure needs to be defined to support a range of projects. An effective structure is one in which the respective roles are clear and communication is facilitated. This can be scaled to each project, from a two-person team to larger. An effective structure has related activities brought together in an organizational entity (team), while facilitating both formal and informal communications between and among all of the entities. Because a system is being developed, the various disciplines that make up these teams, such as hardware, software, or human-machine interface, are not independent of one another. Cross-coordination must be ongoing throughout project development. This means that continuing communication across disciplines is an essential function of the project organization for successful system development.

More specifically, the following paragraphs highlight the key criteria for an effective system management organization as adapted from *Management of Systems Engineering*, by Wilton P. Chase:

Facilitate communications

Few of the problems that arise in developing a system can be solved by a single discipline. Each provides a way of looking at the system, but complete understanding requires integrating these perspectives. This system view is an ongoing need, and so the various team members must coordinate as the system is being developed, understand the viewpoint of the others and communicate in a language understandable to all.

Streamline controls

A clear statement and understanding of the level of detail to be controlled at the project level will make management more efficient. This will keep the managers from slipping into too much detail in the specialties that they came out of. The sections in Chapter 4 that address controls give guidance on how to tailor the process appropriately.

Simplify the paperwork

Standardized documentation is essential for efficient system management, to record and transmit analyses, plans, and designs. During much of the systems engineering process, documentation is the only product. The system design is described only in specifications. The chapters of the Guidebook that follow provide guidelines for developing documents appropriate to the scale and complexity of the project at hand.

Types of organizational structures

Functional One common approach is a functional configuration, in which each functional specialty or discipline is assigned to individual organizational entities. As an example, consider a systems engineering team that performs all systems engineering across all projects. This works best for small projects, in which the team members may be working on several projects at once. Communications problems can occur, though, for larger projects and when subsystem teams are created. The risk is that the subsystem teams optimize for the subsystems and not the system. Also integration may be difficult since the pieces have been developed independently. This means that frequent cross-disciplinary communication and consideration of the system-level issues are essential.

Project The other approach is centered on projects, not disciplines. All those working on a project, no matter what their specialty, will report (possibly indirectly) to the project manager. This works only if the project is so large and long-term that the specialists can devote themselves to it for an extended period.

Matrix A hybrid approach is the matrix management structure, in which team members report to both project and functional management. This is effective for large, long-term projects.

Project Office Another approach is to have project management, systems engineering, and design organized by project and request project support from the functional staff as needed. This works for a moderate sized project, in which only the key individuals devote full time to the project, while the specialists work on multiple projects.

Integrated Product Team (IPT) This is a team consisting of both agency and contractor representatives, working together to develop the system that meets the needs. In a large project there are often mirror functions in the agency and contractor teams; for example, each will have a program manager and a systems engineer. Each will work closely with their counterpart in the IPT. Further, representatives of each of the disciplines will be part of the IPT to ensure the essential cross-discipline communication. Additional IPT's may be formed to address key cross-discipline issues, such as cost of ownership, overall system performance, or configuration management.

Example organizational roles

Figure 3-8 is an example of roles that are generally required for a successful systems engineering organization (adapted from Chase). This may appear frighteningly complex, especially for an agency that typically does small projects. The important thing to keep in mind is that each box represents a role, not a department or even an individual. A simple project may only require two people, a project manager and a systems engineer, with help from administrative functions on an as-needed basis. For larger organizations that manage more complex projects, this is a template for a structure that groups like activities together while maintaining system-level oversight and coordination.

There are three major activities in the organization: project management, systems engineering, and project control. Project management is concerned with planning and execution. Project control tracks the effort relative to its performance, cost, and schedule goals. The same person may assume these two roles. Systems engineering is responsible for design, implementation and verification.

Relationship to consultants and vendors

There is no single correct organizational structure. It needs to be tailored for each team based on existing structures and capabilities within the agency. It should take effective advantage of in-house expertise and existing working relationships and communication paths. There are no standard roles for agencies and contractors. Agencies can and often will develop their own software, for example. Similarly, an agency may choose to outsource oversight activities. The only caveat is that there are certain activities that can only be performed by the agency. These key activities are listed for each step in the process throughout Section 4. The keys to a successful team that includes consultants are appropriate roles and frequent and frank communications.

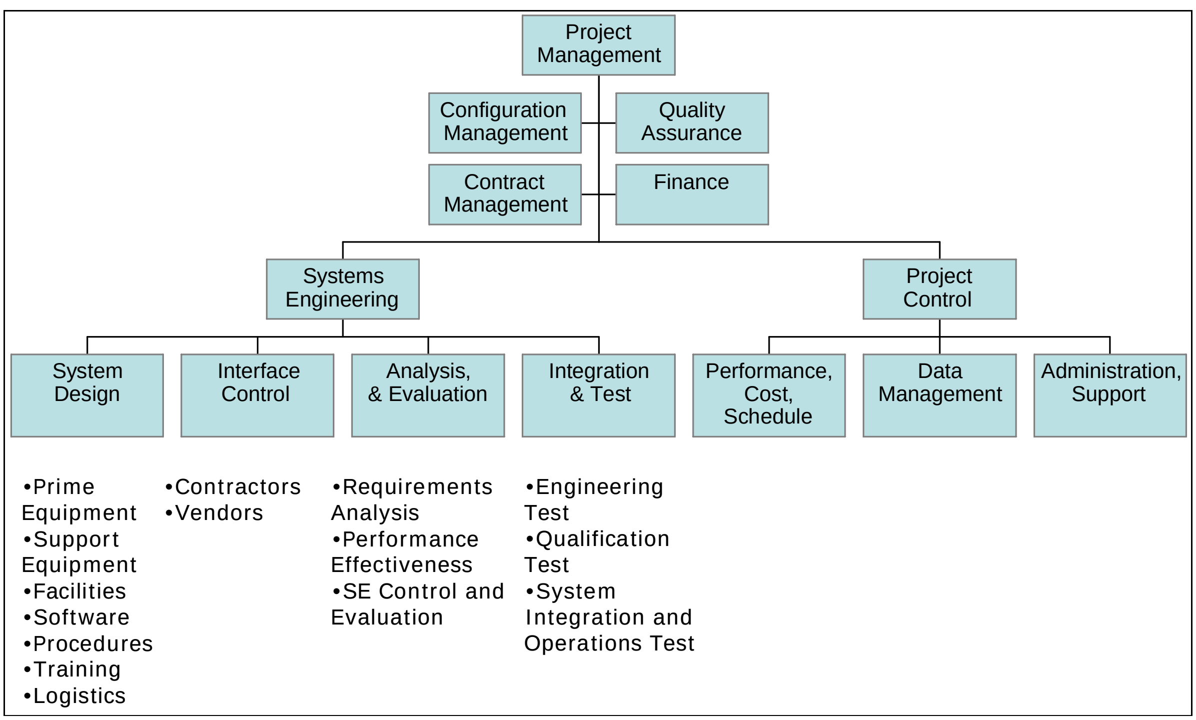


Figure 3-8 Example Organization

3.10 Procurement Options

OBJECTIVE:

Describe procurement options, types and techniques available for the acquisition of Intelligent Transportation Systems (ITS) and some examples of how they can be used.

The following are options that can be used for obtaining services to develop ITS projects. Agencies with an internal pool of technical resources may elect to develop the entire system in-house. Most agencies, though, will use a combination of in-house management and oversight with procurement for development and integration services.

In-house Development

System owners that elect to use the internal resources and capabilities of the organization to perform the development activities should use the processes described in this Guidebook. Internal agreements should be written and signed between the system owner and development teams as though they were procured from the outside. In addition, there should be an independent review (by another division, agency or independent consultant) of the products and activities. Even though the development is done internally, an independent review team is recommended in order to provide a "sanity check" on the development. This will build confidence in the project and help identify and manage project risks.

Contracted Services

The following is a brief description of the two basic classifications of procurement that are common for building transportation capital projects:

- ***Engineering and Design Services:*** In traditional infrastructure construction, this type of procurement is used for planning and the development of Plans Specifications & Cost Estimate (PS&E). The contractor selection for this type of procurement is based on qualifications.
- ***Construction services:*** In traditional infrastructure projects, construction follows PS&E and is the "installation" phase of the project. Construction contractor selection is based on the bid price.

In this Guidebook, reference is made to Consultant, System Manager, Systems Engineering Assistance, System Integrator, Independent Verification and Validation (IV&V). These contracted services are used to carryout various aspects of ITS project development. It is recommended that the ***Engineering and Design Services*** procurement option be used to contract for these services. This allows the agency to select the appropriate team based on their qualifications and not on the lowest price.

Construction services (low bid process) should continue to be used for routine ITS field elements (poles, cabinets, pull-boxes and installation.), building the TMC, or standard items like Model 170 or 2070 controllers with standard modules. The Construction services option is NOT recommended for the other system development services noted above that include specialized hardware and software procurement or development and integration.

Some key procurement issues and techniques related to ITS developments

The following is a brief description of the primary types of contracts that are used in ITS procurements and some relevant issues and techniques associated with each.

Fixed Price – System Owner contracts a single price for all products and services to implement the project. This is sometime referred to as low bid or lump sum.

Fixed Price is usually associated with the low bid used with Construction procurements. This type of contract transfers the project risks to the contractor. If there is a cost overrun, the contractor absorbs this overrun and if they perform better than planned, the contractor's profit is higher. In ITS developments the System Owner who uses a fixed price contract needs to know exactly what is expected and clearly specifies it to the contractor. Standard performance specifications are in place and special provisions documented for the work to be contracted. If not, the contractor can interpret the vague scope of work in their favor in order to meet profit goals. (e.g. reduced documentation, testing, proprietary solution)

Since all risks are absorbed by the contractor, fixed price bid will be higher to reflect this uncertainty.

Cost-reimbursement (Cost plus) – System Owner reimburses the contractor for labor, material, overhead and administration costs plus a fixed fee.

Cost reimbursement type contracts are used where there is a high level of project risk and uncertainty. With this type of contract the risks are primarily on the System Owner. The contractor gets reimbursed for all of their costs. Additional work performed due to changes or rework, the contractor will get paid this additional effort. The overall budget is managed by the System Owner with the responsibility to manage the contractor to that budget. This type of contract is recommended for the system definition and hardware and software development where there is the risk of stakeholder changes to the system.

A variation on this type of contract, which has been used in the past for ITS projects, is a combination of a Cost reimbursable (Cost plus) but with a cost cap on the total project that the contractor cannot exceed and is responsible to manage (contractor has the project risks). This is essentially a fixed price contract. ITS projects are not well defined in the early stages of system definition and there is a great deal of unknowns and risks of stakeholder changes. In these cases this variation on the Cost-reimbursement (Cost plus) option is not recommended.

Time and Materials (T&M) type of contract – System owner pays an hourly rate that includes all profit and overhead, and the materials are billed separately.

This type of contract is very similar to the Cost reimbursement (Cost plus) type of contract except the contractor rolls all labor, overhead and fees into an hourly rate. The system owner only sees this rate. Materials are paid separately.

This type of contract is recommended when the risk of stakeholder changes to the system is high or stakeholder involvement requires unknown number of meetings, reviews, and iterations on definition and design.

Task ordering – This is a technique for managing a project that has a number of tasks but the detailed scope of each are not well specified upfront. This can also apply where the system owner has multiple contractors and consultants under a single contract. This technique allows a great deal of flexibility to the System Owner for systems development. The following are examples of how task ordering can be used for ITS developments.

- Each phase of the project can be executed with a sequence of task orders. For example, the task would be for the development of a concept of operations, or the development of the system requirements. At the end of task the System Owner may elect to issue another task to carry the work forward or use a different consultant or contractor.
- Another example would be for the development of alternate designs from multiple development teams. Each design is evaluated when complete and the best design or combination moves forward for implementation. For example, the National ITS Architecture development was done using four (4) independent teams working concurrently, and at the end of this phase the best aspect of each was integrated together into a single architecture we use today.
- For projects where there is an overlap between a consultant phase and the development team's phase of work, a task order can be used to bring in a development team into the project early. The system owner would get support during the earlier phase activities without being committed to the development team for the next phase of work.

4 ITS Program Lifecycle Framework

OBJECTIVE:

To provide an overview of the Intelligent Transportation lifecycle model including the development process prescribed by this Guidebook. This section describes the ITS program lifecycle and its relationship to Information Technology (IT) and State capital project development lifecycles. It also identifies key phase decision points and the sub-processes within each phase. It briefly describes these sub-process steps and provides a primer for the reader who is not familiar with the systems engineering process.

Section 4 and 4.1 describes the phases, tasks and activities of the ITS Project Lifecycle Framework. The section starts with an introduction to the lifecycle model and the need to successively refine needs, goals and objective over the project lifecycle, a comparison of the lifecycle phases for capital project development projects, Information Technology projects and Intelligent Transportation Systems (ITS) projects. Then a brief description of each step of the lifecycle with crosscutting tasks (4.1.1), finally the introduction concludes with a roadmap of the major sections of the guidebook that can be used to navigate through the ITS project lifecycle tasks described in section 4.

Section 4.2-4-8 describes the individual tasks and activities for each phase of the lifecycle as follows:

Section 4.2 Phase 0 – Concept Exploration and Feasibility: Interfacing with the regional architecture, concept exploration and feasibility assessment.

Section 4.3 Phase 1 – Planning and Concept of Operations: Project planning, systems engineering management planning, and concept of operations.

Section 4.4 Phase 2 – Systems Definition: Requirements development, high level design, and component level detailed design.

Section 4.5 Phase 3 – Systems Development and Implementation: Hardware and software development, integration, verification and initial deployment.

Section 4.6 Phase 4 – Operations and Maintenance: Validation, operations and maintenance, changes and upgrades.

Section 4.7 Phase 5 – Retirement and/or Replacement of the system of major subsystems.

Section 4.8 describes the crosscutting tasks that apply to one or more phases of the project lifecycle: Stakeholder involvement, elicitation, project Management practices, risk management, project metrics, configuration management, process improvement, control gates, trade studies, and technical reviews.

Comparison of the common lifecycle models

Figure 4-1 illustrates a comparison of the lifecycle models. 1) Capital project development 2) Information Technology Systems (email system, intranet, or information management system), 3) Intelligent Transportation System (freeway management system or incident management system), the phases are similar among the three lifecycle models with variations to the tasks and activities performed within each phase which are domain-specific. This Guidebook describes the detailed process steps for Intelligent Transportation Systems. Major phase decision points are noted by the “Stop” signs at these points in the lifecycle a major decision made e.g. the continuation of the project or a major procurement.

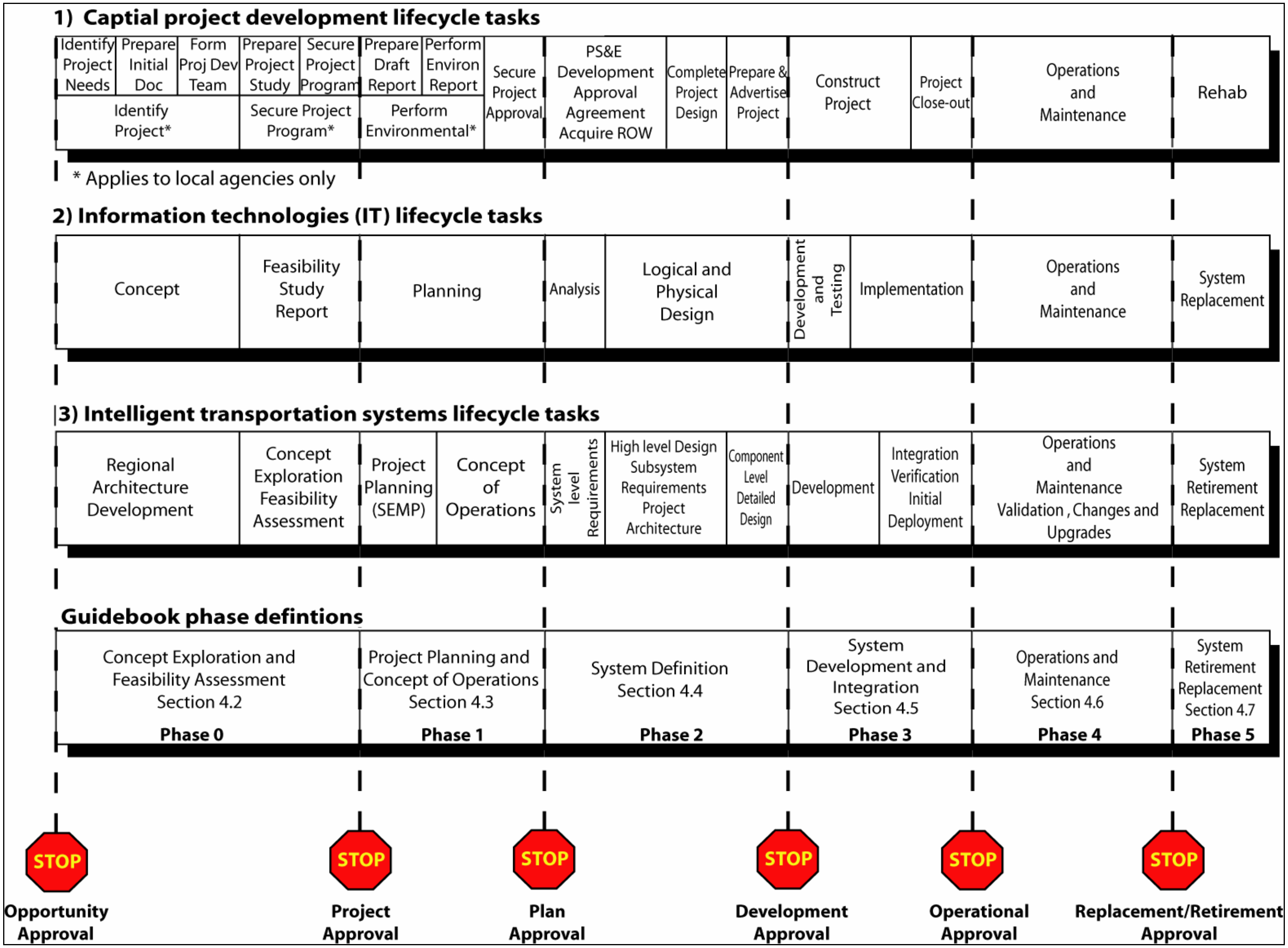


Figure 4-1 ITS Program Lifecycle Framework

4.1 Overview of the Lifecycle Model for Intelligent Transportation Systems

The basic tenets of systems development are continuous refinement and increased definition of the system over time. The figure (Figure 4-2) below illustrates the relationship of each phase of the lifecycle to the detail needed for system definition and the refinement of needs, goals, objectives and expectations. The lifecycle of the system may also be viewed as a spiral where each whorl is an increased level of system definition. The first whorl is used to identify the regional architecture, gather a comprehensive set of needs, goals, objectives, expectations and the candidate set of projects. The next whorl analyzes and prioritizes these items and evaluates alternative solutions and creates the business case through a feasibility assessment for the recommended project. The next whorl (above the Vee) is the development phases of the project. This generates the needed system definition to develop, implement, operate and maintain. Finally these whorls continue throughout the life of the system and represent the upgrades and evolution of the system until retirement.

It is important to use a top down successive refinement of the set of goals, objectives, needs, envisioned solutions and expectations for each phase of the lifecycle for ITS projects for the following reasons:

Whorl 1 – Its purpose is to gather a comprehensive set of goals, objectives, needs, expectations and envisioned solutions:

At the beginning, when the regional architecture is being developed, it is important to be as inclusive as possible as to what the stakeholders desire for the envisioned solutions. This tends to generate a large number of needs at a very high level (user services, market package, major data flows). This ensures that, as much as possible, nothing is missed as the project moves forward.

Whorl 2 – Its purpose is to prioritize and analyze (cost/benefit) the set of potential concepts:

The next level of refinement takes place in the concept exploration and feasibility phase. Analysis is done on alternative concepts. This analysis identifies the relative costs and benefits of the alternative project concepts, and recommends a concept to move forward into development. This analysis refines the envisioned solutions and prioritizes the goals, objectives, needs and expectations. The Stakeholders are involved in the selection of the recommended concept that will be moved forward into development.

Whorl 3 – Its purpose is to build a project that meets stakeholder needs:

The next level of refinement occurs throughout the project development. The Concept of Operations is where the envisioned solution (recommended system concept) is modeled for its operations from multiple stakeholder viewpoints. As a result, the needs, goals and information become very specific. The maintainers, operators, and managers will have very specific needs and specific ideas on the way they would like the system to meet those needs.

Whorls 4 & 5. – Their purpose is to adjust and "fine tune" the system through modifications and upgrades in order to build on the synergy of the system and look for new opportunities.

The final and on-going level of refinement is in the continuous improvement of the features of the system. The existing system provides an opportunity to define new needs based on real world experience in the use of the system and to adapt the system to the changes in the environment and in the stakeholder needs. For example, the changeable message sign system has been adapted to function as an Amber Alert system (new need).

With each phase of the project, the definition of the system should become clearer and there should be a convergence in stakeholder consensus on needs, objectives and priorities. In a multi-regional system, this takes time since concepts such as the sharing of information and control may encounter institutional barriers and a natural resistance to change. Each stakeholder must become comfortable with these concepts and internal policies may need to be changed to support them. This iterative approach enables the stakeholders to identify and address these kinds of issues early and, if some of these concepts cannot be implemented, the stakeholders will understand the constraints before projects are started or defined.

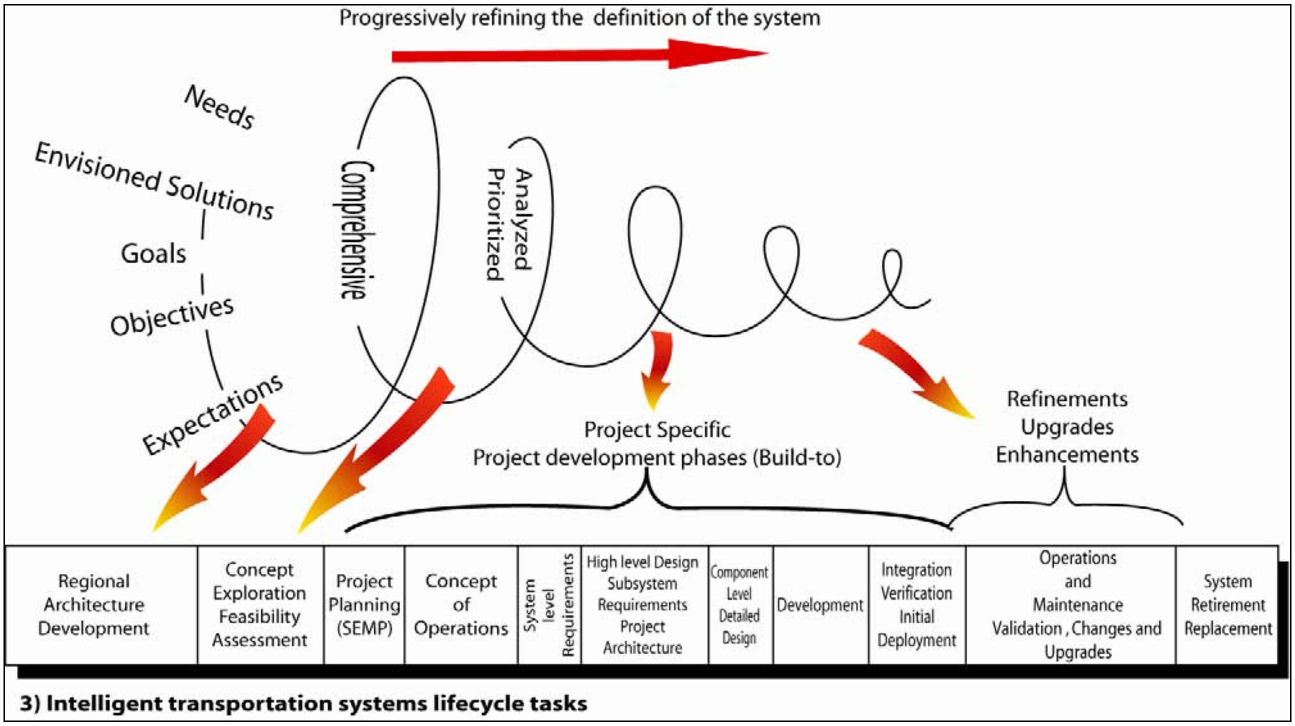


Figure 4-2 Spiral Nature of Systems Development

Figure 4-3 illustrates the relationship of the lifecycle tasks to system development model below by the formation of the Vee diagram encompassing the development phases of the project, starting from project planning Phase 1 to Phase 4. The realization of these relationships became the foundation of the technical development model that is called the Vee technical development model that is described throughout the rest of this guidebook. It is described here to transition from the linear representation to the Vee Development Model used in the remainder of this Guidebook.

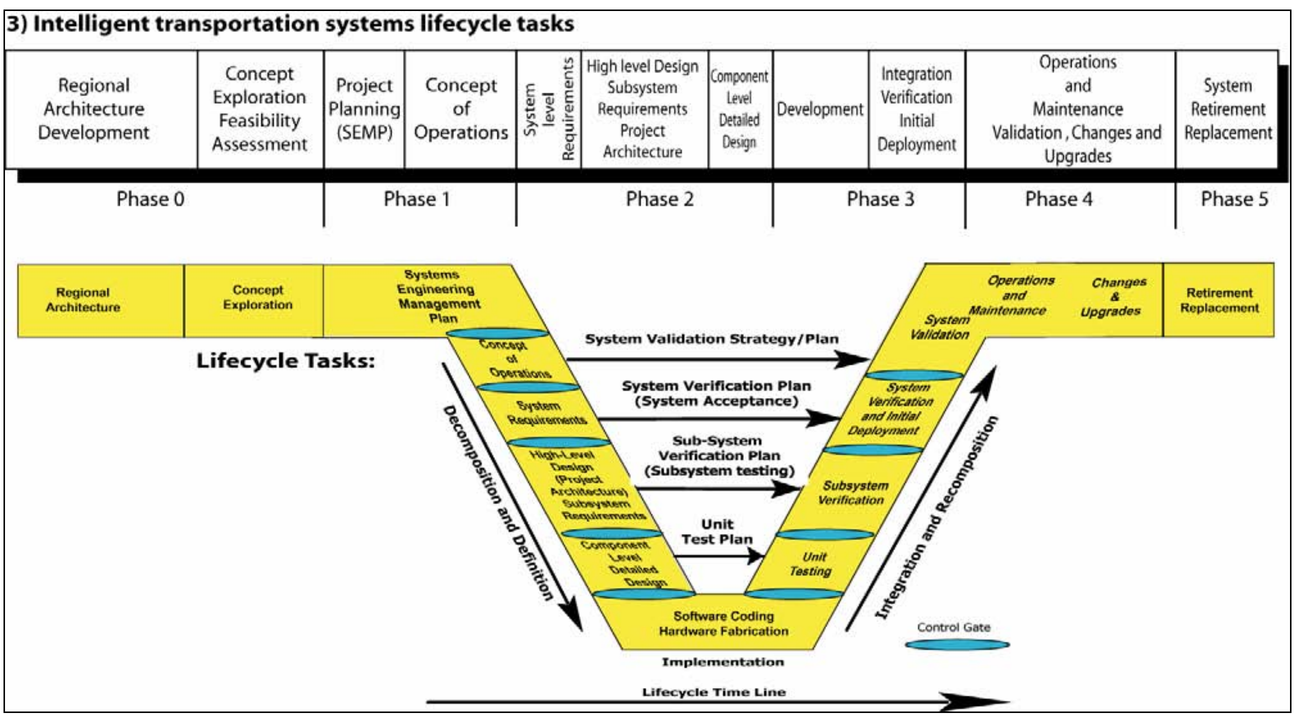


Figure 4-3 Transition from the Linear Systems Lifecycle to the Vee Technical Development Model

4.1.1 Description of the Lifecycle Model

OBJECTIVE:

To provide an overview description of the ITS lifecycle model and the activities associated with each phase. The Vee Development Model addresses the portion of the lifecycle model for system development and implementation. In addition, this section will describe the crosscutting activities that are enablers for the process steps and will provide some basic systems engineering principles, terms and definitions to get the reader started with this section of the Guidebook.

The Vee Development Model is the recommended development model for ITS projects. This model for systems development combines the important features of the classic Waterfall model and the Spiral development model used primarily for software development. Both of these models are briefly described below.

Illustrated in Figure 4-4 is the Vee Development Model in context of the lifecycle framework. This model has gained much acceptance in the systems engineering community and has been illustrated as part of the most recent Systems Engineering Process Standards ISO 15288 and EIA 632 as well as many of the current leading systems engineering texts. The reason for this acceptance is that the model illustrates some key systems principles about the relationship of the early phases of the development to the end results of the project. This is described in more detail in the step-by-step description below. This overview also serves as a primer for the reader who is not familiar with the systems development process.

The following are step-by-step descriptions of the lifecycle model and the crosscutting activities that support the steps of the lifecycle. The title of each section is followed by the number of the section in this Guidebook which contains more descriptive detail. In addition to this description, observations about the Vee Development Model, some basic systems engineering principles, terms and definitions are discussed to give the reader a starting point with this section of the Guidebook. A more comprehensive list of terms and definitions are included in the appendix. The Vee portion of the illustration is the project level development phase. This discussion starts with the description of the left "wing" of the illustration, the Vee technical model itself and finally the right "wing" of the lifecycle framework. It should be noted that the "Changes and Upgrades" step (right "wing") is performed using the Vee technical model but is not illustrated that way for the purposes described below.

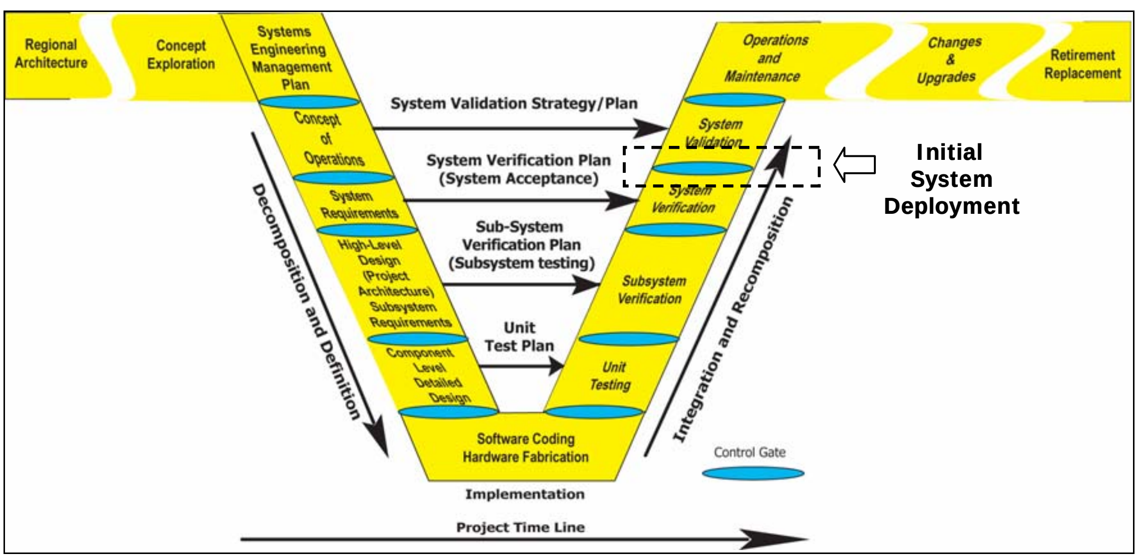


Figure 4-4 Adapted from the Vee Technical Development Model (Forsberg, Mooz, Cotterman INCOSE 1992)

Basic Terms and Definitions

Architecture: *Two definitions –*

1) **Regional** a framework for ensuring technical and institutional integration of ITS systems in a geographic area. For these purposes, a regional (ITS) architecture is based upon the National ITS Architecture tool. 2) **Project** a project-specific description of both logical and physical elements arranged in a hierarchical form and shows interconnections among the elements. It has enough definition that component level detailed design specification can be written and developed.

System is an integrated composite of people, products and processes, which provide a capability to satisfy a stated need or objective.

Systems Engineering is an inter-disciplinary approach and a means to enable the realization of successful systems. Systems engineering requires a broad knowledge, a mindset that keeps the big picture in mind, a facilitator and a skilled conductor of a team.

FHWA Final Rule The FHWA Rule on Architecture Standards and Conformity (Final Rule), also referred to as 23 CFR 940, requires the development of regional ITS architectures (RA's) and that all ITS projects using Federal funds be developed using a systems engineering analysis. The systems engineering analysis includes: identification of the portion of the RA being implemented, participating agencies roles and responsibilities, requirements definition, alternatives analysis, procurement options, identification of applicable ITS standards and testing procedures, and procedures and resources for system operations and management.

Process Activities
Section 4.2.1 to 4.7.1

The following is a summary of the process steps in the Vee technical model.

Interfacing with the regional ITS architecture and planning (4.2.1)

Development of a regional ITS architecture is not covered by this Guidebook and is well described in *"ITS Regional Architecture Guidance Document: Developing, using and maintaining an ITS architecture for your region – October, 2001.* This section does cover the interface between the regional ITS architecture and planning. Two of the key activities of this phase are the identification of the regional stakeholders and the building of consensus for the purposes of information sharing and long term operations and maintenance. The candidate projects are then put into the transportation planning process (TIP, STIP, RTIP). For more information on developing a regional ITS architecture please refer to Regional ITS Architecture Guidance document from the website: http://www.its.dot.gov/arch/arch.htm.

Concept Exploration and Feasibility Assessment (4.2.2 & 4.2.3)

Concept Exploration is used to perform an initial feasibility and benefits analysis and needs assessment for the candidate projects from the regional ITS architecture development. This would result in a feasibility study report and specific cost benefit analyses for alternative project concepts. The output of this stage is a definition of the problem space, key technical metrics and refinements to the needs, goals, objectives and vision. The stage identifies the highest cost/benefit project concept (best business case); the one that should move forward into development. This activity may result in combining or dividing candidate projects based on the best cost/benefit analysis.

Systems Engineering Planning (4.3.1 & 4.3.2)

Each project that moves forward into development must be planned. This planning takes place in two parts. In part one, the system owner develops a set of master plans and schedules that identifies what plans are needed and, at a high level, the schedule for the implementation of the project. This becomes the framework for what is developed in part two. In part two, the plans are completed during the steps from the concept of operations to the high level design. These plans, once approved by the system owner, become the control documents for completion of the development and implementation of the project.

Concept of Operations (4.3.3)

The Concept of Operations is the initial definition of the system. At this stage, the project team documents the way the envisioned system is to operate and how the envisioned system will meet the needs and expectations of the stakeholders. The envisioned operation is defined from multiple viewpoints for example, operators, maintainers, and managers, and how the system will be validated (proof that the envisioned system meets the intended needs). A refinement of the problem space definition, needs, goals, expectations, stakeholder lists, and project constraints is placed into the concept of operations document. This document contains the updated, refined summary of the work done at the concept exploration phase.

System Level Requirements (4.4.1)

Requirements are developed for the system. At the system level, the definition of WHAT the system is to do, HOW WELL it is to do it and under WHAT CONDITIONS are documented. The system requirements are based on the user needs from the Concept of Operations. Requirements do not state HOW (design statements) the system will be implemented unless it is intended to constrain the development team to a specific solution.

High Level Design (Project Architecture) and Sub-system Requirements (4.4.1 and 4.4.2)

The high level design stage defines the project level architecture for the system. The system level requirements are further refined and allocated (assigned) to sub-systems of hardware, software, databases and people.

Requirements for each sub-system element are documented the same way as was done for the system level requirements. This process is repeated until the system is fully defined and decomposed. Each layer will have its own set of interfaces defined. Each layer will require an integration step that is needed when the sub-system is developed. The control gate that is used for this final review is sometimes called the Preliminary Design Review (PDR).

Component Level Detailed Design (4.4.3)

At the component level detailed design step, the development team is defining HOW the system will be built. Each sub-system has been decomposed into components of hardware, software, database elements, firmware and/or processes. For these components, detailed design specialists in the respective fields create documentation ("build-to" specifications) that will be used to build or procure the individual components. A final check is done on the "build–to" specifications before the design moves forward to the actual coding and hardware fabrication. At this level the specific commercial off-the-shelf hardware and software products are specified but they are not purchased until the review is completed and approved by the system owner and stakeholders. The control gate that is used for this final review is sometimes called the Critical Design Review (CDR).

Hardware/Software Procurement or Development (4.5.1)

This stage involves hardware fabrication, software coding, database implementation and procurement and configuration of off-the-shelf products. This stage is primarily the work of the development team. The system owner and stakeholders monitor this process with planned periodic reviews, e.g. code walkthroughs and technical review meetings. Concurrent with this effort, unit test procedures are developed that will be used to demonstrate how the products will meet the detailed design. At the completion of this stage the developed products are ready for unit test.

Unit Testing (4.5.3)

The components from the hardware and software development are verified in accordance with the unit Verification Plan. The purpose of unit test is to verify that the delivered components match the documented component level detailed design. This is done by the development team in preparation for the next level of integration. This is a good review point for the system owner and stakeholders.

Sub-system Integration and Verification (4.5.2, 4.5.3)

At this step, the components are integrated and verified at the lowest level of the sub-systems. The first level of verification is done in accordance with the Verification Plan and is carried out in accordance with the Verification Procedures (step-by-step method for carrying out the verification) developed in this stage. Prior to the actual verification a test readiness review is held to determine the readiness of the sub-systems for verification. When it is determined that verification can proceed, the sub-systems are then verified. When the integration and verification is completed, the next level of sub-system is integrated and verified in the same manner. The process continues until all of the sub-systems are finally integrated and verified.

System Verification (4.5.3)

System verification is done in two parts, the first part is done under a controlled environment (sometimes this is called a "factory test") and the second part is done in the environment in which the system is intended to operate (sometimes called "on-site testing") and is done after initial system deployment. At this stage the system is verified in accordance with the Verification Plan developed as part of the system level requirements done earlier in the development. The system acceptance will continue through the next stage, initial system deployment. The final part of system verification is then completed. A control gate is used for this conditional system acceptance.

Initial System Deployment (4.5.4)

At Initial System Deployment, the system is finally integrated into its intended operational environment. This step may take several weeks to complete to ensure that the system operates satisfactorily long term; this is sometimes called a "system burn-in". Many system issues will surface when the system is operating in the real world environment for an extended period of time. This is due to the uncontrollable nature of inputs to the system, long term "memory" leaks in software coding and race conditions. (unexpected delays between signals) that may only occur under specific and infrequent conditions. Once the system verification is completed, the system is accepted by the system owner and stakeholders and moves into system validation and operations and maintenance phases.

System Validation (4.6.1)

Validating the system is a key activity of the system owner and stakeholders. It is here that they will assess the system's performance against the intended need, goals and expectations as documented in the Concept of Operations and in the Validation Plan. It is important that this validation takes place as early as possible after the acceptance of the system in order to assess the strengths and weaknesses and assess new opportunities. As a result of the validation new needs and requirements may result. This activity does not check on the work of the system integrator or component supplier (that is the role of System Verification) and is performed after the system has been accepted and paid for. As a result of validation new needs and requirements may be identified. This evaluation sets the stage for the next evolution of the system.

Operations and Maintenance (4.6.2)

After the initial deployment and system acceptance, the system moves into the operations and maintenance phase. In this phase, the system will carry out the intended operations for which it was designed. During this phase, routine maintenance is performed as well as staff training. This phase is the longest phase since it will extend through the evolution of the system and end when the system is retired or replaced. This phase may carry on for decades. It is important that there are adequate resources to carry out the needed operations and maintenance activities; otherwise, the life of the system can be significantly shortened due to neglect.

Changes and Upgrades (4.6.3)

During the operations and maintenance phase, if changes and upgrades are needed, it should be done in accordance with the Vee technical process as recommended by this Guidebook. Using the Vee process for changes and upgrades will help maintain system integrity (maintain synchronization between the system components and its respective documentation). Sometimes existing systems (legacy systems) have not been well documented. In such cases, it is recommended to first perform a reverse engineering process on the target areas of proposed change in order to develop the needed documentation for the forward engineering process.

Retirement/Replacement (4.7.1)

At some point in the life of a system, it may be necessary to retire and/or replace the system. The system may no longer be needed, may not be cost effective to operate, may no longer be maintainable due to obsolescence of key system elements or this may be a planned activity where an interim system was put in place for a period of time until the final system was ready for deployment. This stage looks at how to monitor, make the decisions needed and prepare for this event.

Cross-cutting Activities 4.8.1 to 4.8.12

A number of cross cutting activities are needed to support the development of Intelligent Transportation Systems. The following are the essential enabling activities used to support one or more of the lifecycle process steps.

Stakeholder Involvement (4.8.1)

Stakeholder involvement is regarded as one of the most critical enablers within the development and lifecycle of the project and system. Without effective stakeholder involvement, the systems engineering and development team will not gain the insight needed to understand the key issues and needs of the system owner and stakeholders. This will increase the risk of not getting a valid set of requirements to build the system or to get buy-in on changes and upgrades.

Elicitation (4.8.2)

Elicitation is the act of effectively and accurately gathering information needed to develop the system. Needs, goals, objectives, requirements, and other information are obtained by a discovery process. Some of the information is documented or otherwise clearly stated but much is implied or assumed. This enabling process helps draw out and resolve conflicting information, build consensus, document and validate this information.

Project Management Practices (4.8.3)

Various project management practices are needed to support the development of the system. Project management practices provide a supportive environment for the various development activities. It provides the needed resources, then monitors and controls cost, schedules and communicates status between and across the development team members, system owner and stakeholders.

Risk Management (4.8.4)

There will be risks for ITS system development efforts. Risk Management is a process used to identify, analyze, plan, monitor and then to mitigate, avoid, transfer or accept these risks.

Project Metrics (4.8.5)

Project metrics are measures that both the project manager and the systems engineer use to track and monitor the project and the expected technical performance of the systems development effort. The identification and monitoring of metrics are important so that the team can determine if the project is "on-track" both programmatically and technically.

Configuration Management (4.8.6)

Managing change to the system is a key process that occurs throughout the life of the system. Configuration management is the process that supports the establishment of system integrity (the documentation matches the functional and physical attributes of the system) and maintains this integrity throughout the life of the system (synchronizes changes to the system with its documentation). The lack of change management will shorten the life of the system and may prevent a system from being implemented and deployed.

Procurement Options (4.8.7)

Procurement options are important for the system owner and stakeholders. The goal in choosing a procurement option is to give the system owner the greatest flexibility and to manage project risk appropriately. The choice depends on the phase of work being done. Some phases of work will lend themselves better to one type of procurement option over another.

Deliverables/Documentation (4.8.8)

Examples of products are identified as one would expect from each phase of the development and system lifecycle. Asking for the appropriate documentation at the appropriate level of quality will drive the quality of system that will be delivered.

Process Improvement (4.8.9)

A quality aspect of the systems lifecycle is to continuously improve the process and to learn from previous efforts to improve future work that may be done. Process improvement is an enabler that will provide insight on what worked and what needs improvement in the processes. This activity is used to improve the system owner's and development team's documented processes over time.

Control Gates (4.8.10)

Control Gates are formal decision points along the lifecycle that are used by the system owner and stakeholders to determine if the current phase of work has been completed and that the team is ready to move into the next phase of the lifecycle. By setting entrance and exit criteria for each phase of work, the control gates are used to review and accept the work products done for the current phase of work and also evaluate the readiness for moving to the next phase of the project.

Trade Studies (4.8.11)

Technical decisions on alternative solutions are a key enabler for each phase of system development. This starts when alternative concepts are evaluated, and continues as requirements are decomposed and allocated to sub-system developing, the high level design is developed and commercial off the shelf products are assessed. This section provides a method to perform a trade study.

Technical Reviews (4.8.12)

Technical reviews are used to assess the completeness of a product, identify defects in work, and align the team members to a common technical direction. This section provides a process for conducting a technical review.

Key Observations for the Vee Development Model

1. The left side is the definition and decomposition of the system into components that can be built or procured. The bottom of

the Vee is the construction, fabrication and procurement or development of the component items. The right side of the Vee integrates the components into sub-systems and finally into the final system. Each level of integration is verified against the left side of the Vee through the Verification Plans (verification process (4.5.3)).

2. Control gates (4.8.10) provide the system owner with formal decision points to proceed to the next step of the process. A control gate is an interface from one phase of the project to the next and there is an interface between each phase on the left side to the right side.
3. There is a relationship of the activities performed on the left side of the Vee to the products produced, integrated and verified on the right side of the Vee (model versus reality).
4. The view of the system that is most important for the system owner and stakeholders is at the Concept of Operations level. Below this level is the area of most interest to the development team and the area for which they are responsible (system owner responsibility versus the development team responsibility).
5. Importance of stakeholder involvement shows on the left side for defining the system and on the right side for the verification of the system.

Some Basic Systems Engineering Principles

The Systems Engineer must:

1. View the system from the stakeholder points of view – (walk in the shoes of the system owner and stakeholders) – Key processes include needs assessment, elicitation, Concept of Operations and stakeholder involvement.
2. Start at the finish line – define the output of the system and the way the system is going to operate. – Key processes include Concept of Operations and Validation Plan.
3. Address risks as early as possible – where the cost impacts are lowest. – Key processes include risk management, requirements and stakeholder involvement (spend more time on the left side of the Vee)
4. Push technology choices to the last possible moment. Define what is to be done before defining how it is to be done (form follows function).
5. Focus on interfaces of the system and of the project (organizational, teams and process interfaces).
6. Understand the organization of the system owner and stakeholders and understand the organization of the development team.

Overview of Phases, Tasks and Activities

The following five (5) tables (Table 4-1 thru Table 4-5) contain a list of tasks, activities, products and control gates for each phase of the ITS project lifecycle. This is intended to provide a quick reference that can be used with the overview of each step of the process provided in this section.

Roadmap through Section 4 for each phase, tasks and activities

Figure 4-5 is the overall roadmap that will guide the reader through each phase, tasks and activities of the ITS lifecycle. Each phase starts with a phase roadmap into the tasks of the phase. The following is the list of phase roadmaps:

Phase 0: Figure 4-6 Phase 0 - Concept Exploration and Feasibility Assessment Roadmap

Phase 1: Figure 4-7 Phase 1 - Project Planning and Concept of Operations Development Roadmap

Phase 2: Figure 4-8 Phase 2 - System Definition Roadmap

Phase 3: Figure 4-10 Phase 3 - System Development and Implementation Roadmap

Phase 4: Figure 4-13 Phase 4 - Validation, O&M, Changes and Upgrades Roadmap

Phase 5: Figure 4-14 Phase 5 - System Retirement and/or Replacement Roadmap

Table 4-1 Phase 0 Task, Activities, Products, Control Gates

Phases, Tasks and Activities
ITS Project Lifecycle

	Page 1 of 5 Phase 0 Concept Exploration & Feasibility Analysis		
Phase & Task	**4.2.1 Interfacing with Planning & the Regional ITS Architecture**	**4.2.2 Needs Assessment**	**4.2.3 Concept Selection & Feasibility Assessment**
Activities	▪ Identify the portion of the regional ITS architectures to implement ▪ Study relevant regional ITS architectures ▪ Localize regional ITS architectures ▪ Check consistency	▪ Identify stakeholders ▪ Elicit needs ▪ Document needs ▪ Validate needs ▪ Prioritize needs ▪ Perform gap analysis ▪ Compare costs ▪ Select and document key needs ▪ Validate key needs	• Define vision • Define goals & objectives • Identify constraints • Define evaluation criteria • Identify candidate solutions • Identify alternative concepts • Evaluate alternatives • Document results
Products	Project description, including: ▪ Goals & objectives ▪ Portion of regional ITS architecture ▪ Organizational constraints ▪ Compatibility constraints ▪ Stakeholder identification ▪ Market packages	Needs Assessment document	Concept description, including: ▪ Selection rationale ▪ Recommended concept ▪ Feasibility assessment
Control Gate			Project approval

Table 4-2 Phase 1 Task, Activities, Products, Control Gates

Phases, Tasks and Activities
ITS Project Lifecycle

Page 2 of 5

	Phase 1 Project Planning & Concept of Operations Development		
Phase & Task	**4.3.1 Project Planning**	**4.3.2 Systems Engineering Management Planning**	**4.3.3 Concept of Operations**
Activities	▪ Define & budget all project tasks ▪ Identify needed resources ▪ Identify procurement options ▪ Develop project schedule ▪ Prepare Project Plan ▪ Prepare any necessary supporting management plans	▪ Assess project technical tasks ▪ Identify needed Systems Engineering processes (tailoring) & resources ▪ Prepare Systems Engineering Management Plan Framework	• Define project vision, goals & objectives • Develop operational scenarios • Develop & document project Concept of Operations • Develop Validation strategy and plan • Establish configuration management board • Plan operations and maintenance
Products	▪ Project Plan ▪ Supporting management plans ▪ Request for Proposal	▪ Systems Engineering Management Plan Framework ▪ Supporting technical plans ▪ Request for Proposal	▪ Concept of Operations ▪ Validation Plan ▪ Operations and maintenance plan
Control Gate		SEMP Framework	Concept of Operations

Table 4-3 Phase 2 Task, Activities, Products, Control Gates

Phases, Tasks and Activities
ITS Project Lifecycle

	Page 3 of 5 Phase 2 System Definition and Design		
Phase & Task	4.4.1 Requirements Development	4.4.2 High Level Design	4.4.3 Component Detailed Design
Activities	▪ Develop requirements ▪ Write and document requirements ▪ Check completeness ▪ Analyze, refine & decompose requirements ▪ Validate requirements ▪ Develop verification plan ▪ Manage requirements ▪ Perform Requirements walkthrough ▪ Update Plans ▪ Perform risk management	▪ Develop, decompose & evaluate architecture alternatives ▪ Identify & evaluate internal & external interfaces ▪ Evaluate industry standards ▪ Develop sub-system verification plans ▪ Select & document the high level design ▪ Perform Preliminary Design Review ▪ Identify configuration items ▪ Update SEMP ▪ Perform risk management	• Evaluate commercial off-the-shelf products & applications • Perform detailed design • Perform technical reviews • Develop unit verification plans • Perform Critical Design Review • Identify configuration items • Perform prototyping and modeling • Update SEMP • Perform risk management
Products	▪ System Requirements document ▪ Verification Plan (system level) ▪ Traceability matrix	▪ High Level Design ▪ Interface descriptions (internal and external) ▪ Required industry standards ▪ Sub-system Requirements ▪ Verification Plans (sub-system level) ▪ Updated traceability matrix ▪ Integration plan	▪ List of selected commercial off-the-shelf products and applications ▪ Component Detailed Design ▪ Verification Plan (unit level) ▪ Updated traceability matrix ▪ Updated integration plan ▪ Updated development plan
Control Gate	System Requirements baseline approval System requirements review	Sub-system requirements baseline approval Project Architecture Preliminary Design Review	Component level detailed design Baseline approval Critical Design Review

Table 4-4 Phase 3 Task, Activities, Products, Control Gates

Phases, Tasks and Activities
ITS Project Lifecycle

	Page 4 of 5 Phase 3 System Development & Implementation			
Phase & Task	**4.5.1 Hardware / Software Development**	**4.5.2 Integration**	**4.5.3 Verification**	**4.5.4 Initial System Deployment**
Activities	▪ Support, monitor & review development ▪ Develop system products ▪ Coordinate concurrent development activities ▪ Procure commercial off-the-shelf products ▪ Perform risk & developmental configuration management	▪ Update integration plans ▪ Update integration activities ▪ Perform integration activities ▪ Update deployment plans ▪ Update operations and maintenance plans ▪ Perform risk & configuration management	▪ Update verification activities in SEMP / Project Plan ▪ Develop Verification Procedures ▪ Perform verification ▪ Document verification activities ▪ Verification Readiness Review ▪ Perform risk & Configuration management	▪ Update deployment strategy ▪ Perform deployment activities ▪ Perform system burn-in ▪ Perform risk & configuration management
Products	▪ Hardware components ▪ Software components ▪ Support products ▪ Unit verification procedure ▪ CM reports	▪ Updated integration plans ▪ Integrated sub-system and system ▪ Updated deployment plans ▪ Updated Operations and maintenance plans	▪ Verification Procedures ▪ Verification Reports ▪ Verified sub-systems and system	▪ Deployed System ▪ System acceptance report
Control Gate			Acceptance of sub-systems	Operational baseline approval System acceptance

Table 4-5 Phase 4 & 5 Task, Activities, Products, Control Gates

Phases, Tasks and Activities
ITS Project Lifecycle

Phase & Task	Page 5 of 5 Phase 4 Operations & Maintenance			Phase 5 System Retirement / Replacement
	4.6.1 System Validation	4.6.2 Operations & Maintenance	4.6.3 Changes & Upgrades	4.7.1 System Retirement / Replacement
Activities	▪ Evaluate and Validate system ▪ Document strengths and weaknesses	▪ Update operations & maintenance plans ▪ Collect operations & maintenance information ▪ Perform operations & maintenance	▪ Analyze needed changes & upgrades ▪ Reverse engineering ▪ Forward engineering ▪ Perform configuration managment	▪ Plan retirement / replacement ▪ Perform gap analysis: legacy system capabilities vs. needed system capabilities ▪ Evaluate cost of upgrade vs. replacement ▪ Develop replacement / retirement strategy
Products	▪ Validation Report	▪ Operations & Maintenance Procedures	▪ Legacy system documents ▪ System improvement descriptions ▪ Update system documentation	▪ Retirement / Replacement Plan
Control Gate				Retirement / replacement approval

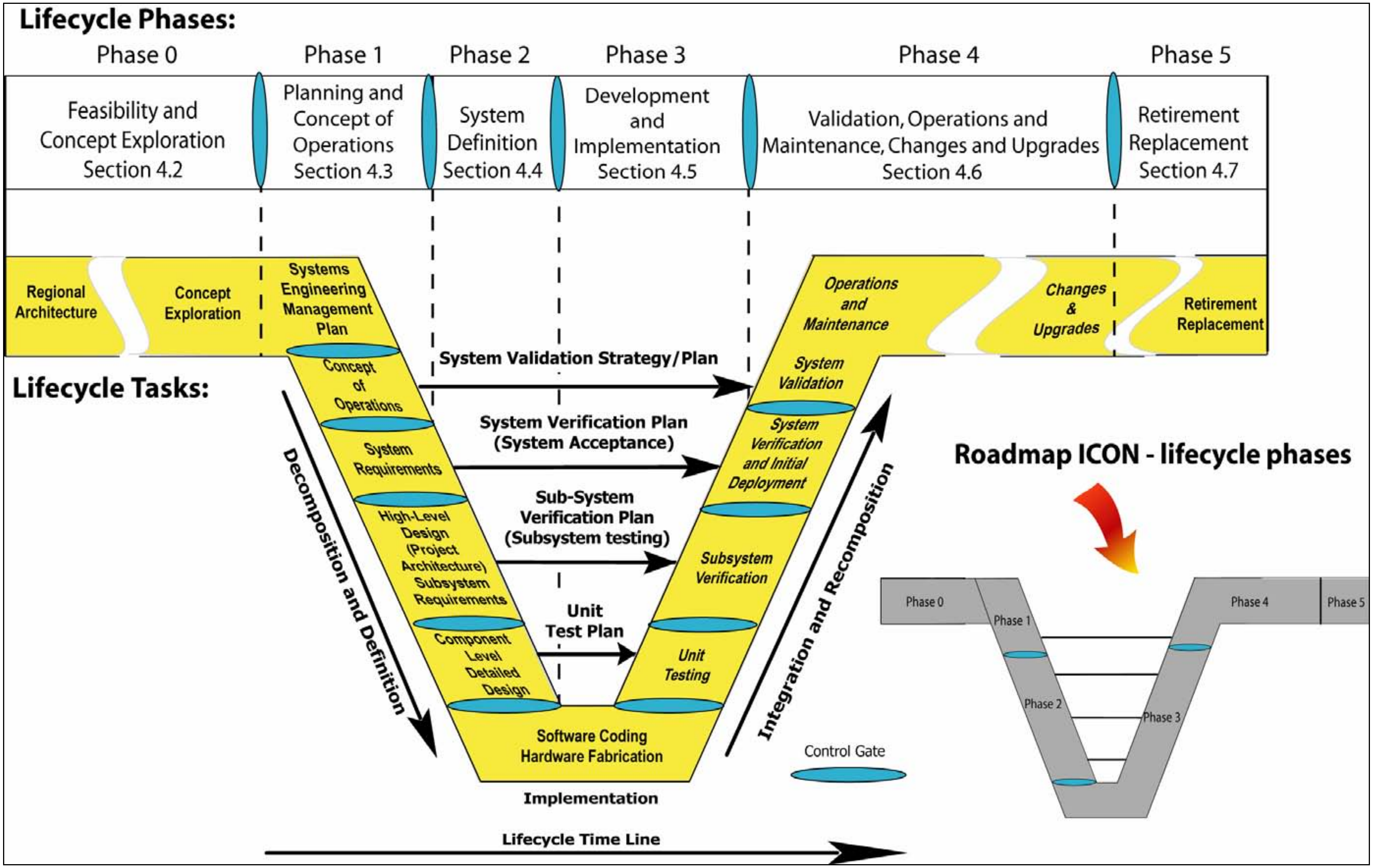


Figure 4-5 Roadmap through Section 4 of the Guidebook

4.2 Concept Exploration and Feasibility Assessment

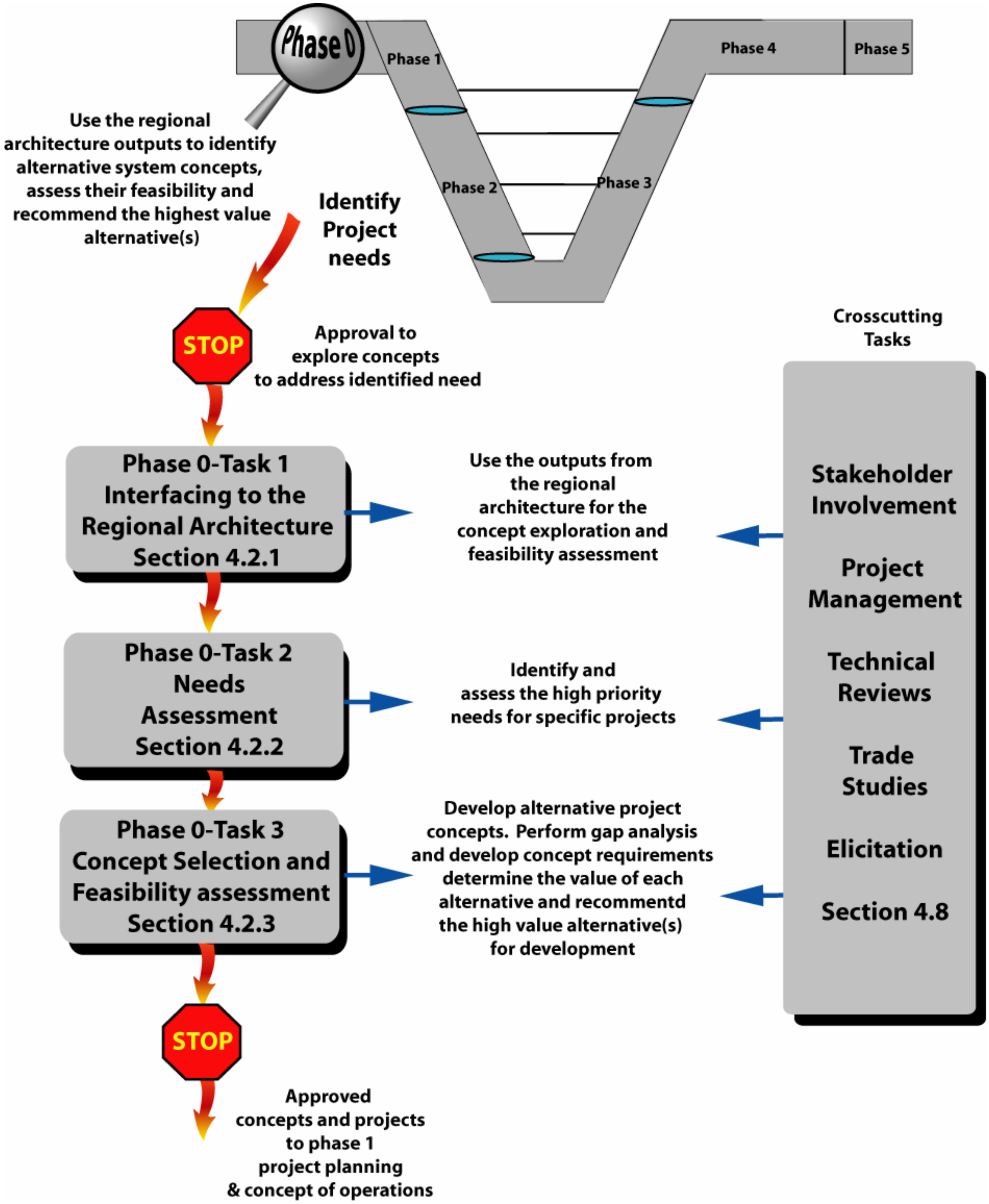


Figure 4-6 Phase 0 - Concept Exploration and Feasibility Assessment Roadmap

4.2.1 Interfacing with Planning and the Regional ITS Architecture

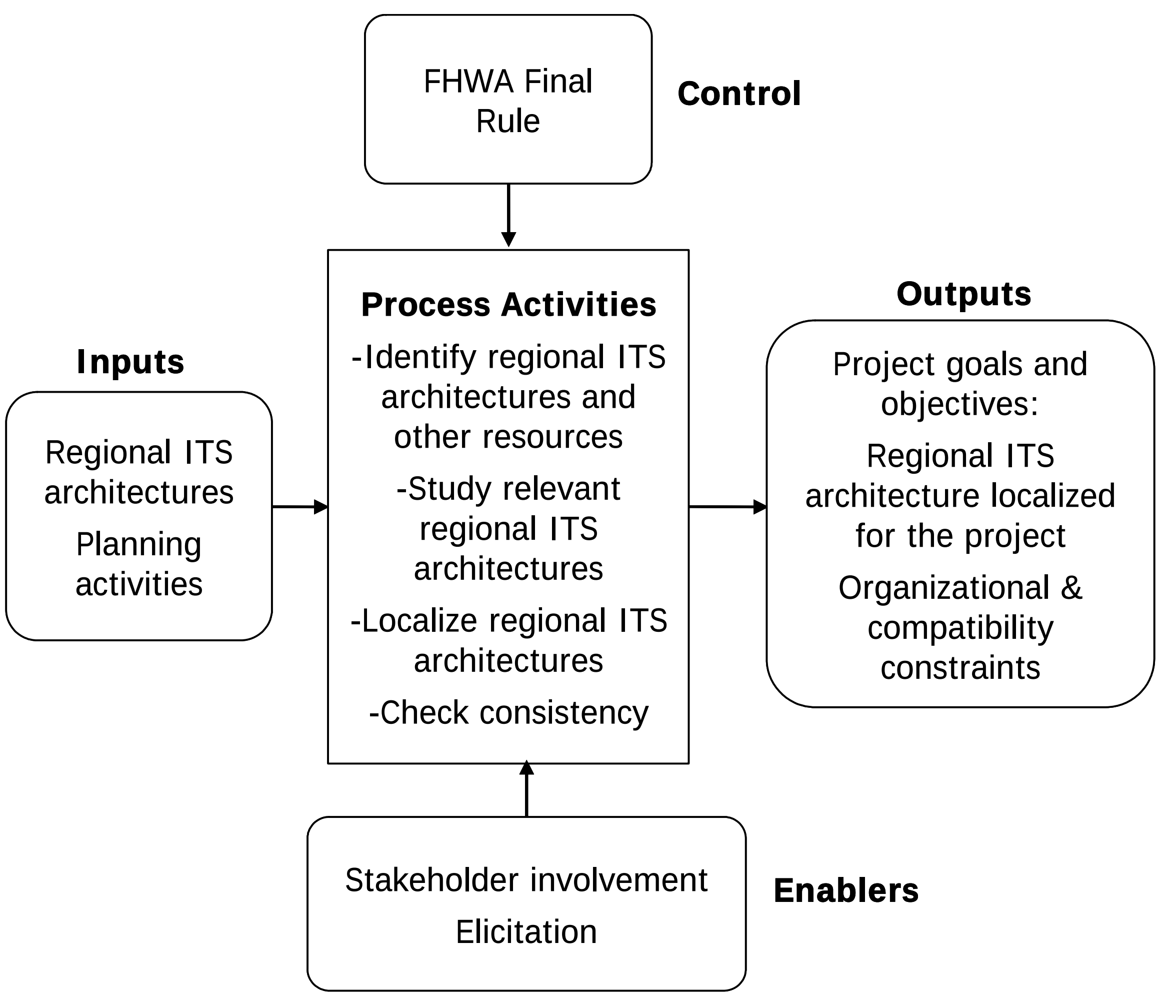
OBJECTIVE:

Intelligent Transportation Systems at the project level are to be consistent with, and leverage from, the regional ITS architecture and planning. This step describes what to expect from the regional ITS architecture and how to use the products at the project level. An existing regional ITS architecture provides products that can be leveraged for concept exploration, feasibility analysis and the project level developments.

DESCRIPTION:

Before the project level development begins, groundwork is laid in the development of a regional ITS architecture and in the planning process. Activities done at the regional ITS architecture level produce a set of products that include: the inventory of existing systems and processes, identification of stakeholders, identified user needs, regional vision, high-level operational concept, high level functional requirements, and conceptual recognition of regional system interconnections and information exchanges. The candidate project concept will need to refine and expand products from the regional ITS architecture, for example, to expand the stakeholders to include maintenance, IT, and operators that may not have been considered at the regional level. In turn, additional needs may be identified that were not envisioned at the regional level. Interface with planning is essential since it will be a planning role to include ITS projects in the various statewide plans.

CONTEXT OF PROCESS



PROCESS FOR INTERFACING WITH PLANNING AND THE REGIONAL ITS ARCHITECTURE

Inputs:

Regional ITS architectures describe the framework for integration. The project must fit into these architectures (or the architectures must be changed to reflect new regional consensus).

Planning activities provide guidance for the project.

Control:

FHWA Final Rule specifies requirements for receiving federal funds, including the use of the National ITS Architecture and development of and compatibility with a regional ITS architecture.

Enablers:

Stakeholder involvement focuses the project on local needs.

Elicitation draws out and clarifies local project needs.

Outputs:

Project goals and objectives identify what the system is being designed to do and what it will accomplish.

Regional ITS architecture, localized for the project, identifies the portions of the regional ITS architecture selected for development on this project.

Organizational and compatibility constraints come from the organizational and physical context from the surrounding areas and are imposed by the regional ITS architecture.

Process Activities:

Identify existing regional ITS architectures and other resources from the planning process

Many states and regions have developed or are developing state and regional ITS architectures. These architectures provide a good starting point for the project being developed. As of April 8, 2005, your project is required to accommodate a regional ITS architecture. In a very large region, such as a state, or a very large metropolitan area like San Francisco, the architecture may not have sufficient detail for a small local-level project. In that case, if available, seek out county or sub-regional ITS architectures. These are not required by FHWA Final Rule, but prove to be very useful for local coordination, and are kept consistent with the state architecture. Coordinate with Planning to take advantage of all previous work that they have done. The architecture provides a good starting point for concept exploration. State and regional ITS architectures are built on the National ITS Architecture, and so provide a link to it. This means that basing a project on a state or regional ITS architecture will facilitate compatibility with the National ITS Architecture.

Study relevant regional ITS architectures

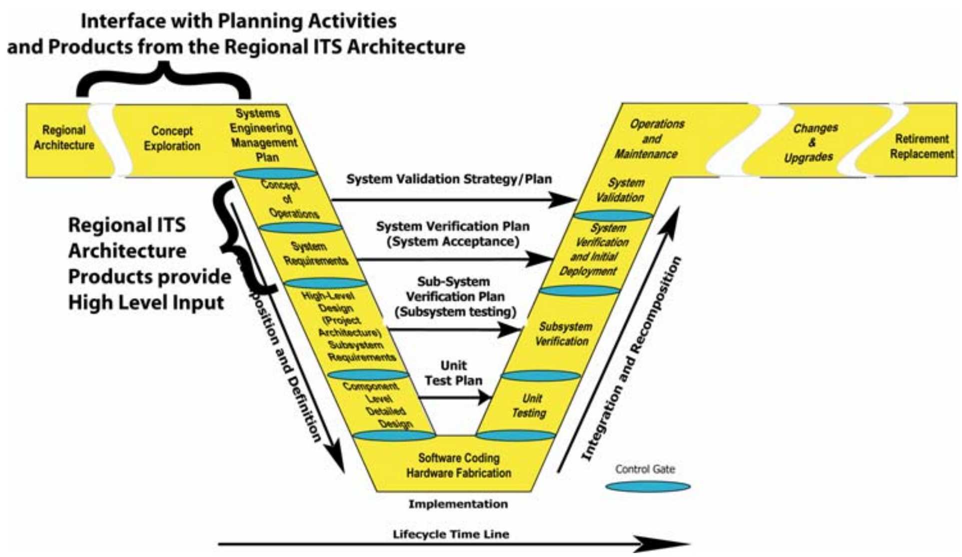
Using a regional ITS architecture will provide a project that is consistent with other systems in the area, meets requirements for federal funding, can be developed more efficiently and quickly and enjoys economies of scale with the surrounding areas. A good regional ITS architecture will provide region-level information in areas of concern in the project development (see checklist).

Localize regional ITS architectures

The regional ITS architecture necessarily addresses some issues that are outside the scope of your project. For example, in a simple signal system that does not interface with ramp meters, the aspects of the regional ITS architecture addressing freeways are not relevant. Also focus on the geographic area for your project. So the first step is to identify the aspects that apply to your project. Then determine what is described by the regional ITS architecture and what will need to be defined.

Check consistency

Confirm that there is nothing planned in your project that would be counter to the regional ITS architecture. Document any constraints that the architecture may place on the design, for example, for compatibility with neighboring systems. This activity will need to be repeated as the concept and design are developed. As of April 8, 2005 the project will be required to be consistent with the regional ITS architecture. This implies a need for regular consistency checks. However, note that the Final Rule allows regional ITS architectures to be modified to accommodate projects (CFR 940.11 d).



Where does interfacing with Planning and the regional ITS architecture take place in the project timeline?

Is there a policy or standard that talk about Planning or the regional ITS architecture?

FHWA Final Rule requires a regional ITS architecture for any region currently implementing or planning ITS projects. All ITS projects must adhere to this regional ITS architecture, although there is a provision for the architecture to be modified to accommodate a project. The FHWA Final Rule also requires that development of a regional ITS architecture be consistent with statewide and metropolitan planning processes.

Which activities are critical for the system owner to do?

- Coordinate with Planning
- Identify applicable regional ITS architectures
- Ensure that project goals and objectives are sufficiently clear to support subsequent tasks

How do I fit these activities to my project? (Tailoring)

The level of activity depends on how many applicable states and regional ITS architectures border your particular region. The more there are, the more work will be expended in this activity to address those external interfaces, but with greater savings in work later, by utilizing the groundwork done by the regional ITS architectures.

What should I track in this process step to reduce project risks and get what is expected? (Metrics)

On the technical side:

- Potential inconsistencies between the regional ITS architecture and the project

Are all the bases covered? (Checklist)

☑ Have all applicable regional ITS architectures been identified?

☑ Have all applicable resources from the planning process been identified?

☑ Has the planned development been checked against the regional ITS architecture to avoid consistency problems during development?

☑ Have all the project-applicable portions of the regional ITS architecture been utilized?

* Required by the FHWA Final Rule - Sec. 940.9

- ☑ Architecture scope (geographic region*, timeframe, range of services, institutions and jurisdictions)
- ☑ Stakeholder identification* (name, responsibility and jurisdiction), range of roles and active involvement.
- ☑ System inventory (all relevant ITS systems existing or planned in the region, with their owning/operating agency and terminators for potential links outside the region; mapping to the National ITS Architecture)
- ☑ Needs and services (description of regional needs and how the needs were determined; existing and planned services, and how these services map to the project concepts called Market Packages in the National ITS Architecture.)
- ☑ Operational concept* (roles and responsibilities of the primary stakeholders and the systems they operate)

- ☑ Functional requirements* (high level functional requirements for each regionally significant system; high level description of what each ITS element will do)
- ☑ Interfaces/flows* (connections between the various ITS systems in the region and the information exchanged)
- ☑ Implementation plan (ties into the transportation planning process)
- ☑ Maintenance plan (in response to changing regional needs and the development of project architectures)
- ☑ Agreements* (a list of the agreements that should be established)
- ☑ Standards identification* (choices of standards among the options consistent with the National ITS Architecture; standards use strategy)
- ☑ Project sequencing* (the order or timeframe in which the projects will be implemented)

Are there any recommendations that can help?

The regional ITS architecture is often developed using the *Turbo Architecture* tool, which structures the information and provides a link with the National ITS Architecture. This provides useful information, including the physical architecture; interconnect diagrams, organizational architecture, data flow diagrams, standards and links to market packages and development activities. It also provides a mechanism for documenting operational concept and high-level functional requirements.

TIP

Several States have developed a statewide ITS architecture with the intent of developing and integrating regional ITS architectures. They are focusing on interregional aspects, including state level services, such as commercial vehicle operations, or services that benefit from interregional coordination, such as trip planning. The goal is to complement the activities of the metropolitan planning organizations by creating a framework for connections between regions and state-level services. A keystone of the process is consensus building among a large, diverse group of stakeholders representing the varied interests throughout the state.

Challenges to traditional planning and ITS project developments at the State level

Interface with planning is essential since it will be a planning role to include ITS projects in the Transportation Improvement Plan (TIP). These projects need to be viable to make it into the planning process and to mainstream Intelligent Transportation Systems projects. This means that the interface point with planning (entry of the project into the TIP) may occur after one of the following events: 1) projects are identified after the architecture development or 2) after the concept exploration and feasibility analysis when the best cost/benefit concept is justified. Both interface points will have their own unique challenges.

At the state level, two different approval and planning paths in developing Intelligent Transportation Systems may occur. The traditional planning and approval process leading to a TIP exists for allocating funding for implementing both the TMC sub-systems and software and for field devices such as changeable message signs, ramp metering, monitoring elements, video monitoring, and fiber optic communications systems. For the field devices to be implemented, the planning process is all that is needed and, once approved, the ITS projects with field elements get deployed. Although the TMC sub-systems have been through the same planning process, these sub-systems are required by some states to pass through a second approval gate controlled by an IT Department, or another branch of state government. In California this is called a Feasibility Study Report (FSR) process which is a requirement of the Department of Finance. Since the traditional planning process (TIP) may take anywhere from one to three years or more, the TMC sub-system elements can be delayed an additional one to three years, with the possibility that the FSR process may cancel the TMC sub-system element while the field elements are implemented.

The first challenge is to integrate the two processes together. This can be accomplished by the use of the concept selection and feasibility assessment defined in section 4.2.3.

The second challenge is to prevent the "shelf-life" of ITS projects expiring due to changing needs and the rapid changes in technology.

4.2.2 Needs Assessment

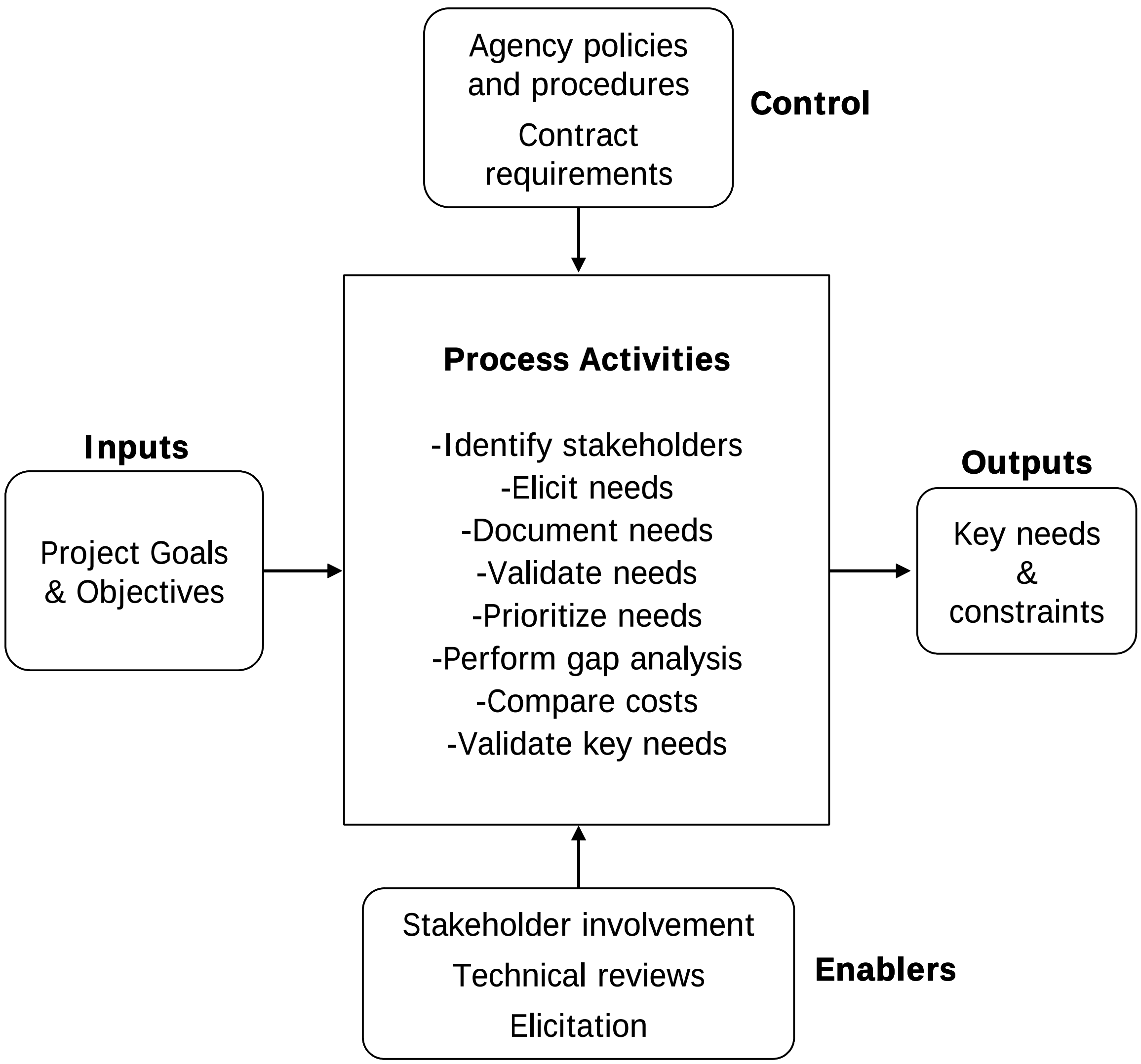
OBJECTIVE:

Needs assessment is an activity done early in system development to ensure that the system will meet the most important needs of all of the people for whom it is being designed, that is, the project's stakeholders. The goal is to ensure that the needs are well understood before starting development. In many cases there will be more needs than can be met, or even conflicting needs, and so prioritization is necessary.

DESCRIPTION:

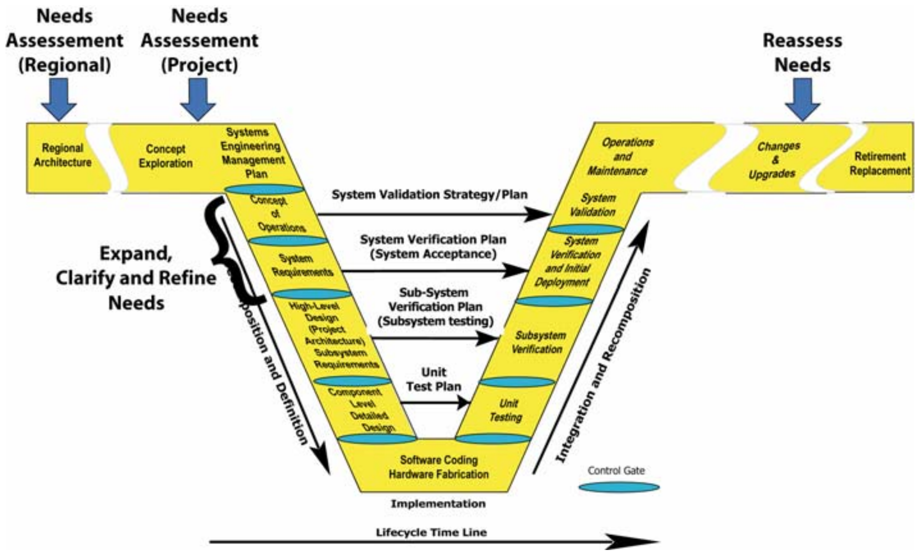
The figure illustrates the needs assessment process. The key is to involve the stakeholders, collect needs from a variety of sources, make sure the needs are well understood, balance and prioritize the needs and document the rationale. This process is done at the beginning of the project and revisited throughout the development. This ensures that the project meets the most critical stakeholder requirements.

CONTEXT OF PROCESS:



NEEDS ASSESSMENT PROCESS

Inputs: *Project Goals and Objectives* are the major drivers for defining the needs. This is an output of the planning process (4.2.1). *Previous studies*, including feasibility studies and strategic plans, are good sources for documented needs.
Control: *Agency policies and procedures* will constrain the process to meet its legal, risk and institutional obligations.
Enablers: *Stakeholder involvement* is essential to defining valid and meaningful needs. *Technical reviews* are an effective means to get stakeholder feedback on the needs being collected. *Elicitation* uses various techniques to draw out, clarify and prioritize needs. *Trade studies* provide an analytical basis for the prioritization of needs.
Outputs: *Key needs and constraints* the list of collected needs, their sources, and documentation of the rationale for the selection of the key needs and any constraints that exist that may limit possible solutions to the needs. This may be a separate document, or incorporated as part of the Concept of Operations.
Process Activities: *Identify stakeholders* Identify the stakeholders who will own, operate, maintain, use, interface with, benefit from or otherwise be affected by the system. *Elicit needs* Needs assessment must set aside any preconceived notions of what the system will do. It then elicits the stakeholders' needs, desires and constraints by various means, as described in 4.8.2. Some of the techniques are literature search, day-in-the-life studies, surveys, one-on-one interviews and workshops. *Document needs* Consolidate the results of the elicitation process into a document. If there are many stakeholders it may be helpful to summarize the results, e.g., 75% of the local agencies cited a need for real-time freeway speed data. Be sure to include all constraints such as restrictions on data sharing. *Validate needs* Present the consolidated results to the stakeholders. This is best done in a workshop where the stakeholders are encouraged to give feedback and have discussions. Continue until they agree that their needs have been captured. *Prioritize needs* Since generally the needs cannot all be met, and sometimes even may be conflicting, analysis of the needs identifies the highest priority ones on which to focus. This may be done by a priorities analysis, surveys or consensus. *Perform gap analysis* Inventory current systems that may contribute to fulfilling the identified needs. Rank each need in terms of both the breadth (e.g., 70% of the freeways currently collect speed data) and depth (criticality) of the gap between current and desired capabilities. *Compare costs* Estimate the cost to meet each of the needs. Qualitative estimates may be sufficient, such as high/medium/low or easy/moderate/difficult to implement. *Validate key needs* Taking into account the priorities, gaps and costs, identify the most pressing needs. Document them and the rationale behind them. Present these conclusions to the stakeholders for discussion and concurrence. Modify key needs as warranted. Update the documentation.



Where does the Needs Assessment take place in the project timeline?

Is there a policy or standard that talks about Needs Assessment?

FHWA Final Rule does not specifically mention general Need assessment practices to be followed. However gathering and assessing needs is an essential part of developing a set of valid requirements, which is required by the FHWA Final Rule.

Which activities are critical for the system owner to do?

- Provide initial statement of needs.
- Provide data and information on current system capabilities relative to the needs.
- Supply any existing documentation of needs.
- Identify the stakeholders and encourage their inputs.
- Participate in any interviews, surveys, workshops or other activities developed for the identification, clarification and prioritization of needs.
- Review statements of needs.

How do I fit these activities to my project? (Tailoring)

These activities are especially important when there are multiple agencies involved, especially if they have different priorities or have not worked together previously. In that case it is essential to get documented agreements on the direction in order to prevent future contention. The larger the number of agencies involved, the more risk there is for conflicting needs, incompatible operations or for the number of needs to become excessive. Hence, the amount of effort expended on needs assessment and prioritization should grow with the number of agencies. On the other hand, a single agency project based on well-defined and limited needs does not need to do any of the prioritizing activities and a one-page needs statement is sufficient. This is the case for most local projects, such as a signal system.

What should I track in this process step to reduce project risks and get what is expected? (Metrics)

<u>*On the technical side:*</u>

- Level of disagreement among stakeholders on high priority needs, since it risks producing a system whose purpose is unfocused and so satisfies no one.
- Percentage of the important needs that cannot be met within the budget, since such needs will drive scope creep.
- Number of expressed needs that are in conflict, since they must be resolved before proceeding.

<u>*On the project management side:*</u>

- Number of stakeholders whose needs have been captured
- Number of stakeholders who agree with the final selection of key needs

Are all the bases covered? (Checklist)

☑ Have all relevant stakeholders been represented?

☑ Have all appropriate resources been utilized to elicit needs?

☑ Have all collected needs and conclusions been reviewed with the stakeholders?

☑ Is there an objective, justifiable approach for prioritizing needs?

☑ Are conclusions and rationale well documented?

☑ Have all stakeholders agreed that their needs are clearly and fairly represented?

Are there any recommendations that can help?

Getting the needs right up front prevents expensive backtracking later on, when changes are much more expensive.

There are *professional facilitators* who can come in to encourage people to work together and to explore new ideas. This might be helpful if there are multiple agencies involved in the project with conflicting needs. There are also techniques that help to draw out, organize, and analyze needs.

Be sure to capture the constraints as well as the needs. A constraint for a single stakeholder, such as the maximum height of maintenance's bucket trucks, will impact the system for all. State policy needs to be considered here. For example, if it prohibits installing private utility lines longitudinally in freeway right-of-way, that will constrain the possible approaches. Be sure the constraints flow into the requirements.

A closer look at Prioritizing needs – Prioritizing needs early is important to prevent making hard decisions later on when it is discovered that not all of the needs can be met within the budget and schedule. Each stakeholder will have their own favorites and you must be sensitive to this and balance the desires of all stakeholders. One way to do this is through an objective priority analysis in which it is clear that all stakeholder needs were given fair consideration. One technique is to draw needs out of previous project documents and prioritize them with concurrence of the stakeholders. Another is a workshop in which stakeholders review and rank candidate needs. Surveys may also be used. All of these techniques are discussed in Section 4.8.11, under the heading, "Making qualitative measures quantitative."

Gap analysis, once the needs have been determined, looks at the gap between current capabilities and the needs. This technique makes qualitative judgments numerical so that they can be compared. Projects are seldom built as a completely stand-alone system but rely on and are built upon legacy systems.

The first step is to determine how far the current capabilities are from meeting the needs, because of insufficient functionality, capabilities, performance or capacity. This is the "depth" of the gap. It may be qualitatively assessed on a scale of 0 (the need is completely met) to 10 (there is no capability currently).

The next step is to determine whether the need is met in some places and not others. This often happens when developing a regional system by integrating local systems, For example, in one study it was found that 70% of the freeway lane miles were instrumented to collect traffic speeds, one of the high priority needs. So in that case there was a 30% geographic gap. This is called the "breadth" of the gap, and is measured as the percentage not covered. The third step multiplies these two metrics giving a unit-less metric that is an indication of how severe the gap is for each need.

Comparing costs is difficult to do at this point, since there is not even a conceptual design. In fact, any cost estimates done this early will rely on assumptions that will certainly change as the project takes form, making the estimates nearly meaningless. However, prioritizing needs should somehow take into consideration the cost of meeting each of the needs. This allows you to leverage a limited budget by possibly addressing many moderately high priority needs, rather than one that may be overly ambitious.

One stand-in for cost is the difficulty of implementing. This can be a qualitative estimate, such as difficult, moderate or easy. What these categories mean can be tailored for your project. For example, an advanced traveler information system had needs for collecting and disseminating various types of travel and traffic status data. The categories chosen were "information that we already collect at a central point," "information that we collect partially," and "information that we do not collect." The first category will be less expensive to implement since only dissemination needs to be done. Now you can look at the prioritized needs relative to their cost categories. Plot the choices' priorities or gaps versus cost, and choose those closest to the easy-important corner.

4.2.3 Concept Selection and Feasibility Assessment

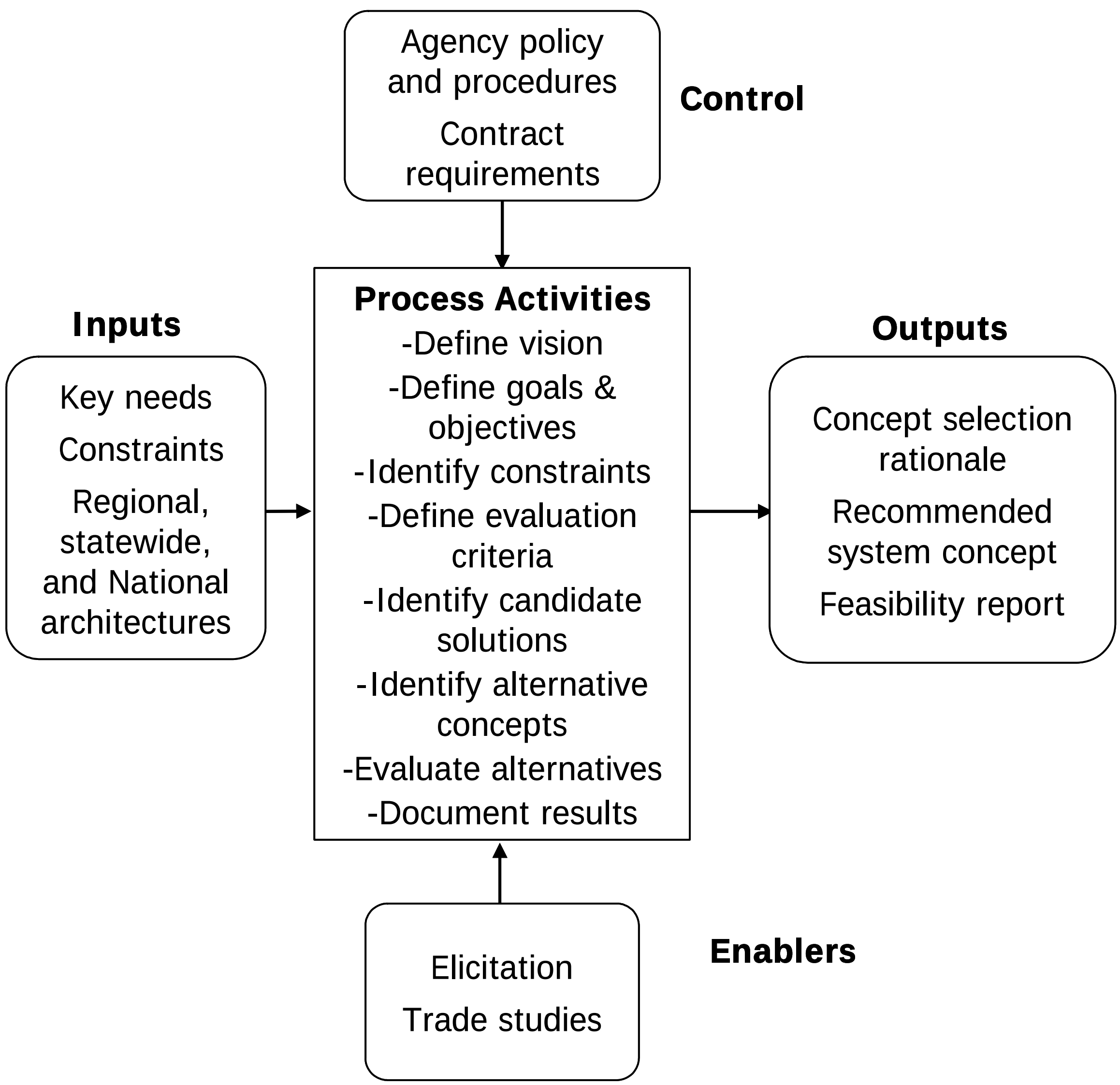
OBJECTIVE:

Concept selection supports Phase 0 to identify promising and feasible projects for development. This activity assesses the best system alternative to implement based on cost and benefit.

DESCRIPTION:

The figure illustrates the Phase 0 steps leading to the definition of a project concept. This is the first step toward developing requirements. The goal is to describe the concept with enough concreteness to develop the concept of operations and to provide something tangible for stakeholder review. This is the bridge between needs and requirements. It is important to satisfy the stakeholders and development team that the selected solution is superior to all other alternatives, in order to start the development going in the right direction. The process is driven by project vision, goals, objectives and constraints. It starts by collecting a broad and varied range of potential approaches to meeting the goals and putting them together into candidate system concepts. These are compared relative to the goals, objectives and constraints. The recommendations provide a documented rationale for the shape the project will take and verification that it is feasible.

CONTEXT OF PROCESS:



CONCEPT SELECTION AND FEASIBILITY ASSESSMENT PROCESS

Inputs:

Key needs come from the needs assessment and identify the transportation needs that indicate a requirement for a project.

Constraints also come from the needs assessment and identify limitations on the design and operation of the system.

Regional ITS Architectures, which may include statewide (inter-regional), sub-regional or county-level, and the *National ITS Architecture* provide guidance and context for the project concept.

Control:

Agency policy and procedures for the procuring agency will constrain the project. State and Federal policies may also influence choices.

Enablers:

Elicitation helps stakeholders provide essential inputs and review.

Trade studies compare alternative concepts.

Stakeholder involvement ensures that the concept meets essential needs without violating any constraints.

Outputs:

Concept selection rationale documents the effectiveness and feasibility of the recommended project concept including justification for the choice in terms of benefit and cost.

Recommended system concept describes the concept selected for best benefit for the cost.

Feasibility assessment or FSR is the document that collects the recommendations and rationale. Your agency may require a formal document in a specified form, such as California's Feasibility Study Report (FSR)

Process Activities:

Each of the following steps is reviewed by the stakeholders.

Define vision

Write one paragraph describing in non-technical terms what the system will do. The idea is to allow lots of stakeholders to review it quickly.

Define goals and objectives

Describe what the potential project should accomplish, from the point of view of the traveling public, the operating agencies and their operators and other stakeholders.

Identity constraints

The constraints come from the regional architecture and inputs from the stakeholders (see Needs Assessment). They will be used to determine feasibility. Constraints may include technical, organizational, funding, schedule, legal and other considerations.

Define evaluation criteria

Evaluation criteria derive from the goals and objectives and are the measures of effectiveness used to compare alternatives. Examples are response time for incident management and average system-wide speeds for a signal system.

Identify candidate solutions

Create a toolkit of technologies and procedures that may help meet the goals. The regional ITS architecture often provides ideas.

Identify alternative concepts

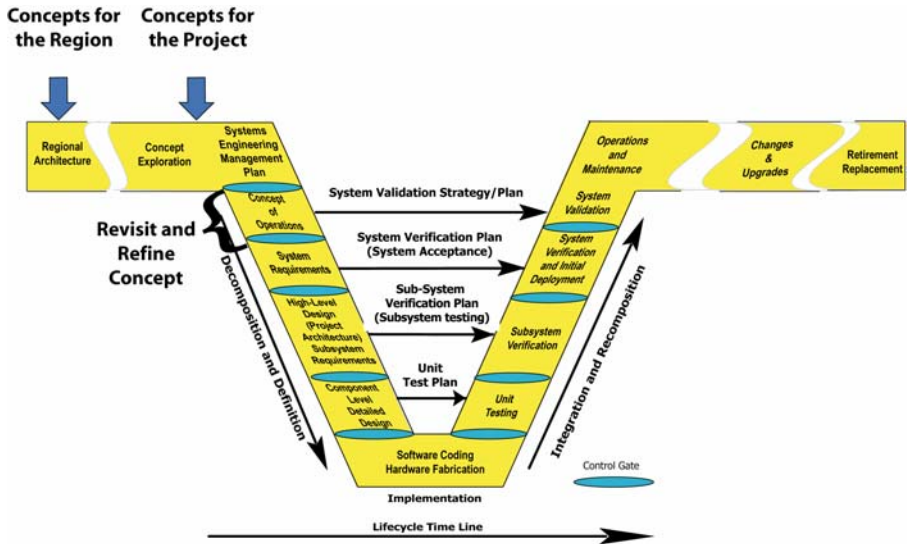
Build project concepts from the candidate solutions or select pieces from the regional ITS architecture. Consider several alternative concepts, ranging from doing nothing to an ambitious system. You may consider both centralized and distributed alternatives. Keep this at a very high level.

Evaluate alternatives

Evaluate benefits, cost, and gaps and compare them as described in Trade Studies and relative to the constraints.

Document results

Document conclusions and rationale in a report. Caltrans includes this analysis in a Feasibility Study Report (FSR).



Where do Concept Selection and Feasibility Assessment take place in the project timeline?

Is there a policy or standard that talks about Concept Selection and Feasibility Assessment?

FHWA Final Rule (23 CFR 940.11) requires identifying the portion of the regional ITS architecture being implemented, identifying participating agencies, defining requirements and analyzing alternatives.

Some states have documented requirements specifically for IT projects. In California, SAM 4819.35 (6/03) requires an FSR for all state IT projects except those with low costs or for acquiring microcomputer commodities.

Which activities are critical for the system owner to do?

- Describe needs, vision, goals, objectives and constraints
- Suggest or review evaluation criteria
- Review candidate concepts
- Review the selection process and conclusions
- Approve the selected concept

How do I fit these activities to my project? (Tailoring)

The level of each activity should be appropriately scaled to the size of the project and the newness of the needs. A small project, or one similar to existing projects, could use a qualitative comparison, a limited number of alternatives and a report of only a few pages. If the operational system will be significantly different from the one it replaces, if it will depend on operational changes or increased inter-agency coordination or if it will address a new set of needs, the alternatives need to be explored in more detail.

This activity may also be dictated by state or regional reporting requirements, such as an FSR. For example, FSR must be approved before State TMC project funding in California.

What should I track in this process step to reduce project risks and get what is expected? (Metrics)

On the technical side:

- Selected Measures of Effectiveness of the system (project-specific) will be used to compare alternatives

On the project management side:

- Number of candidate solutions
- Number of alternative concepts
- Percentage of candidate concepts evaluated
- Percentage of stakeholders who have approved the study

Are all the bases covered? (Checklist)

☑ Is there a validated statement of vision, goals and objectives?

☑ Have constraints been collected from all key stakeholders?

☑ Have the evaluation criteria to be used in comparing alternatives been selected, validated and documented?

☑ Is there a comprehensive list of candidate solutions, both technical and procedural?

☑ Is there a comprehensive and varied list of alternative concepts?

☑ Have you included doing nothing as one of your alternatives?

☑ Has the comparison approach been documented and validated?

- ☑ Is the selected concept and the rationale for its selection documented and has it been reviewed by the stakeholders?
- ☑ Does the documentation satisfy relevant reporting standards, if any, for example, for a Feasibility Study Report if required by the state?
- ☑ Do the conclusions and recommendations flow in a clear and defensible manner from the needs, alternatives selection and analysis?

Are there any recommendations that can help?

Stakeholder involvement is essential at this point to translate needs into requirements. Be sure that the views of operators, owners, maintainers, managers, the traveling public, and other stakeholders are included.

Why are you developing a conceptual architecture this early? Isn't this getting into design?

There needs to be enough specificity to start designing the system. Here it is done at a very high level. For example, you may need to decide whether the system is distributed or centralized. This will make a difference in how the system will be used. The Concept of Operations cannot be written until this is resolved. In other cases there may be multiple ways to meet a need. For example, before designing a bridge across a river, you would need to verify that a bridge is a more cost-effective approach than expanding the existing ferry service.

You will see these same steps used in the design process but at a much more detailed level. At this point the concepts should be developed in no more detail than is necessary to provide a structure for the Concept of Operations. The concept is a tool to gather a complete set of needs and expectations from the stakeholders. It will be successively defined in increasing detail, as discussed in 4.1.1.

A closer look at identifying candidate solutions is key to making sure that you have looked at all of the best approaches. The candidate solutions are your toolkit of technologies and off-the-shelf sub-systems and procedures that will help you achieve your goals. Generally, none of these will solve the whole problem but will address one aspect of it. Examples of candidate solutions are detectors, controllers, workstations, software, communications, vehicles and procedures.

First review all relevant literature, search the web, and query your main customer, your colleagues and technology experts. Brainstorm around each need. Ask yourself what procedures or technologies could help meet the need. Describe each potential solution at a high operational level, for example, a detector that can provide traffic speeds or vehicle-to roadside communication.

Using information gathered from above, construct a straw man list of alternatives, pros and cons of each, needs satisfied by each and query all stakeholder groups, asking if they think each list is complete. Ask if they have anything to add, modify or suggest.

Calculate a rough lifecycle cost, risk or other relevant drawback for each alternative (such as political issues, time to implement or manpower required). Modify the choices where appropriate, possibly changing some alternatives.

Developing alternative concepts comes by synthesizing the candidate solutions into complete systems that work together to meet at least some of the needs. Be sure your list includes a broad range of approaches, including at least the following:

- *Do nothing* This is your comparison case, the choice of just leaving everything as is. You need to convince yourself that the project will generate benefit commensurate with its costs.
- *Do everything* This is the high-end system.
- *Simple and cheap* This is the cost-conscious system, possibly an evolutionary step toward a later "do everything" system.
- *Single need* Focus on the one most essential need.
- *Centralized* Operate from a central point.
- *Distributed* Operate from local points that coordinate.
- *Procedural* Solve the problem without technology, such as by regulations.

4.3 Project Planning and Concept of Operations Development

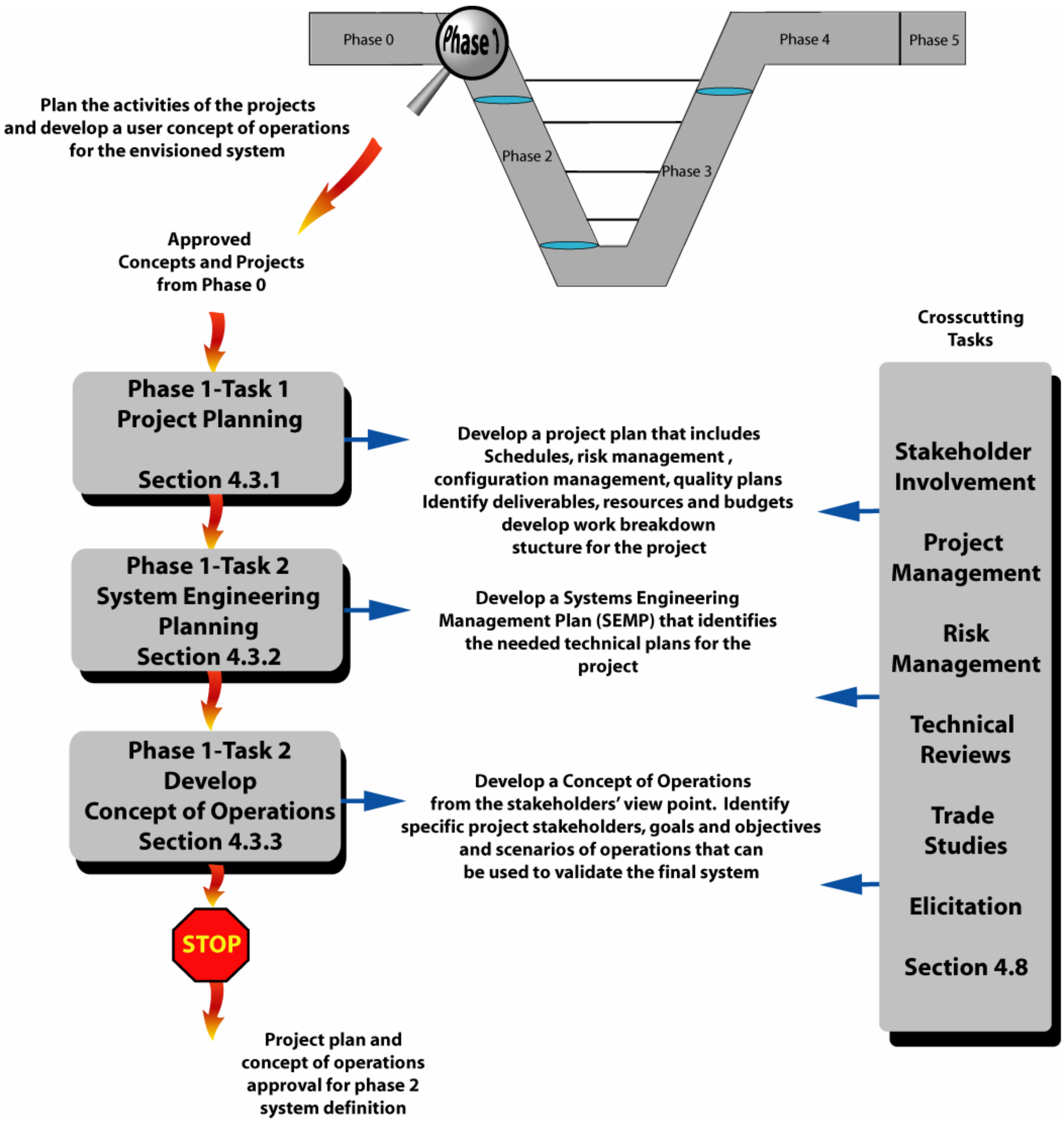


Figure 4-7 Phase 1 - Project Planning and Concept of Operations Development Roadmap

4.3.1 Project Planning

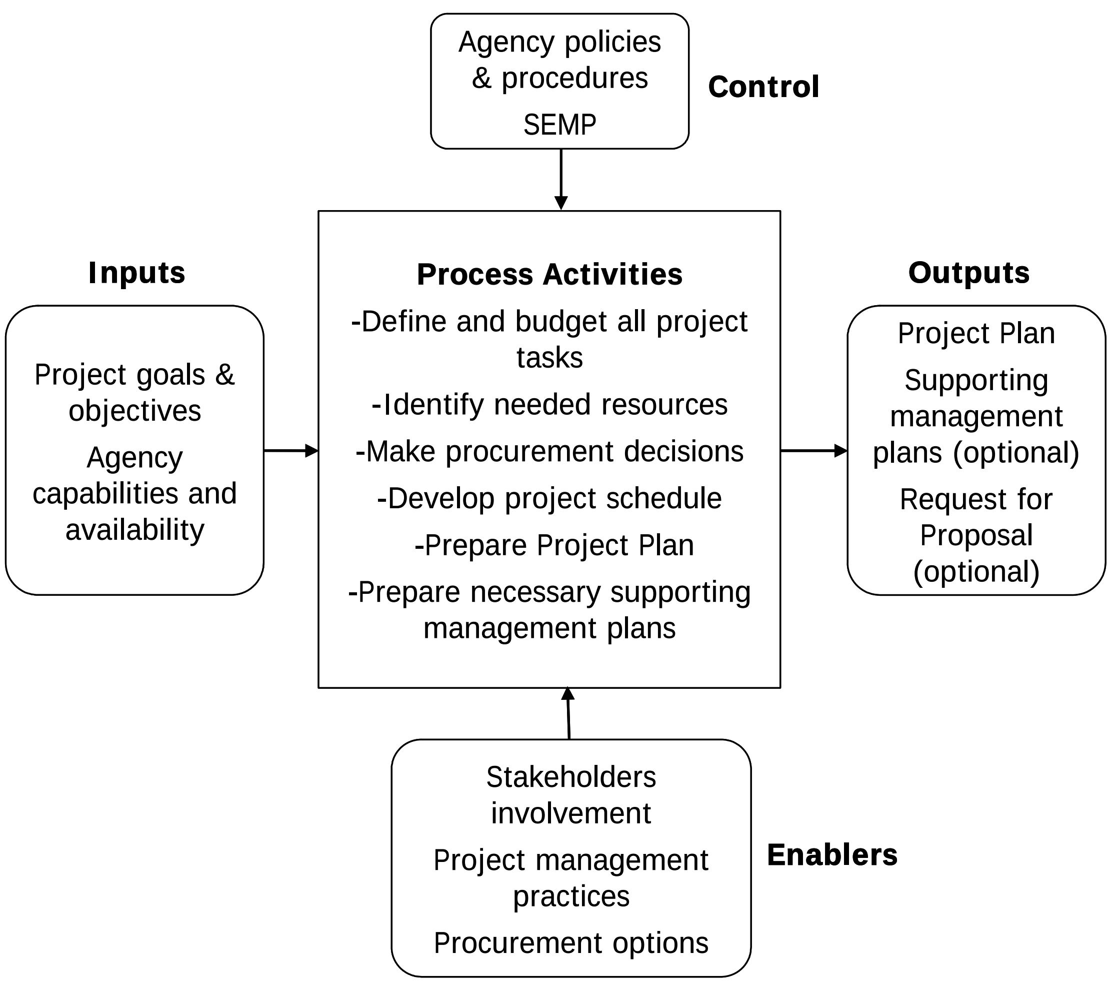
OBJECTIVE:

Project planning identifies the project's needs and constraints at the project level and lays out the activities, resources, budget and timeline for the project. It is an important process because it helps build consensus for the project with the involved stakeholders.

DESCRIPTION:

Project planning starts with the project's goals and objectives as defined by the planning activity, the regional ITS architecture and the needs and constraints elicited from the project's stakeholders. It identifies all relevant agency policies and procedures on managing and executing such a project. It uses these to identify the project tasks (both administrative and technical), their interdependencies, estimates of needed resources and budget for each task, the project schedule and the project's risks. The result of this planning is the Project Plan. This plan identifies the detailed work plans for both the administrative and technical tasks. The plan estimates the resources (people, equipment, and facilities.) needed for each task along with an estimated budget for each task. It identifies key events and the technical and program milestones and establishes a schedule for the project. Each task's detailed work plan is developed to identify its needed inputs and outputs and a description of the process used to carry out the activity. Based on project complexity, additional technical plans (e.g., a Systems Engineering Management Plan) and additional administrative plans (e.g., Configuration Management, Risk Management and Procurement) may be needed.

CONTEXT OF PROCESS:



PROJECT PLANNING PROCESS

Inputs:

Project goals and objectives are defined by Planning, by the regional ITS architecture, and by collected stakeholder needs and constraints

Agency capabilities and availability are the basis for decisions on whether to perform any of the project's tasks in-house or to contract out the effort to either a commercial firm or another agency

Control:

Agency policies and procedures are acknowledged and provide guidelines on how the project is to be managed

SEMP establishes a high level description of the systems engineering effort needed for development

Enablers:

Stakeholder involvement is needed to obtain support for project activities

Project management practices as routinely practiced by the system owner are the basis for project planning

Procurement options will be analyzed and a procurement method selected for any project task that will be contracted out

Outputs:

Project plan establishes a description (what is to be done, what funds are available, when it will be done and by whom) of the entire set of tasks that the project requires

Supporting Management Plans (optional) are needed to provide additional details about any task or group of tasks

Request for Proposal (optional) will be needed for any contract effort

Process Activities:

Define and budget all project tasks:

The first task in planning your project is to identify and define all of the work efforts or tasks, which are needed to accomplish your project's goals. These tasks include, but are certainly not limited to: project management itself and other administrative tasks e.g., financial administration and contract support) some of which may provided by other departments in your agency. The Project Plan also must identify the technical tasks, including the necessary systems engineering activities as described in this Guidebook. Optionally, these activities may be elaborated in the Systems Engineering Management Plan (SEMP).

Identify needed resources:

As part of the planning process, the resources needed for each task must be identified and obtained. Initially, this involves selecting a staff of agency people to manage the project, including selecting a project manager. This also may involve recruiting new people into your organization. Other resources, such as a testing laboratory, may not be needed immediately, but the need for them should be identified as soon as possible. The time-phased staffing plan also needs to consider agency staff to supervise contractors and consultants.

Make Procurement decisions:

Often, some of the project tasks will be contracted out. Aside from any necessary hardware procurement, many of the systems engineering tasks may be best served by commercial firms.

Develop project schedule:

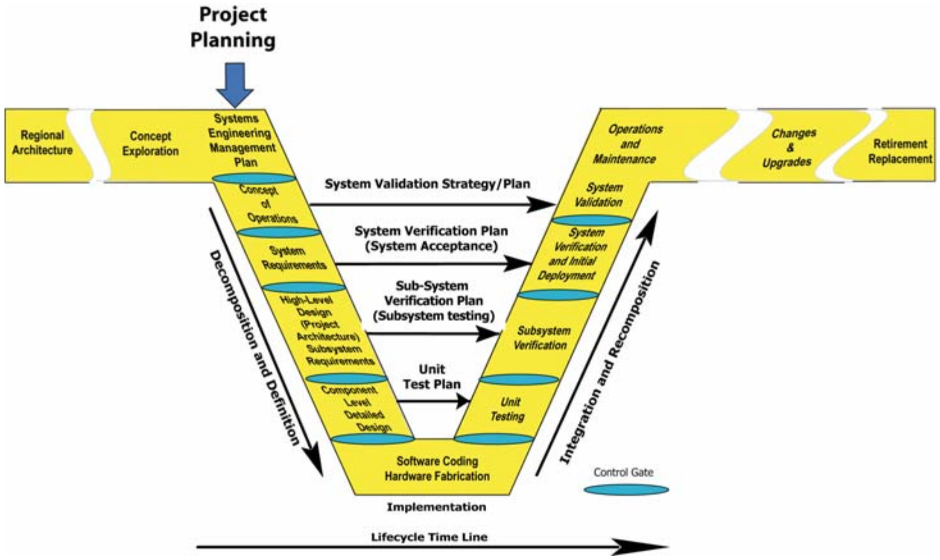
An understanding of the project's tasks, and the resources and budget needed for each task, are combined into a project schedule. This schedule is generally constrained by external requirements, such as, a need for the system to be operational by a certain date or a dependence on installation of another interfacing system.

Prepare Project Plan

The various parts of the project plan need to be gathered together into a written Project Plan. The degree to which the Project Plan needs to be documented will vary by project size and complexity.

Prepare necessary supporting management plans:

Some projects may warrant preparation of separate plans for a variety of specific project tasks and supporting activities. Many of the processes described in this Guidebook have technical planning documents associated with them like an Integration Plan, a Verification Plan or a Deployment Plan.



Where does Project Planning take place in the project timeline?

Is there a policy or standard that talks about Project Planning?

Of all the processes described in this Guidebook, project management planning is the one that is most likely to be defined and controlled by established agency procedures. Almost all agencies have internal rules, regulations and guidelines for project management activities. Further, in the area of procurement, project management intersects with contract law and becomes subject to legal requirements. It is the task of project management to be aware of, use and be compliant with this guidance.

Which activities are critical for the system owner to do?

Of all of the processes of this Guidebook, this one falls most heavily on the system owner, who is most accountable for the project's success. The activities include:

- Ensure that the project's tasks, budget and schedule are necessary and sufficient to support the project's objectives.
- Obtain the necessary resources (people, facilities and intra- and interagency support)
- Establish the means (processes, products, budget and schedule) by which each participant contributor's effort can be measured.

How do I fit this step to my project? (Tailoring)

The degree to which the various management plans are documented is the prime variable in this process step. They must be documented enough so that the responsible staff knows what to do (the larger the staff, the more important this is). On the other hand, they must not be documented in so much detail that they will be ignored by the same staff in order to get the job done on time and within budget. For small and less complicated projects, a single document (the Project Plan) can contain all the information necessary. Many tasks, even technical tasks, which the organization routinely does, will have their own procedures and processes, which the Project Plan can reference. If the project includes custom software development, a SEMP is likely necessary. In addition, the system owner must have available a Configuration Management (CM) Plan designed for software products. The system owner must ensure the organization's standard CM Plan is sufficient, and if not, tailor it to the project or have one prepared.

What should I track to reduce project risk and to get what is expected? (Metrics)

Task budget and expenditure

Task schedule and performance

Task deliverables

Are all the bases covered? (Checklist)

☑ Has an effective project manager been selected?

☑ Have all project tasks been identified?

☑ Have all project tasks been defined enough so they are understood by the performing organization?

☑ Does the performing organization agree the task budget is sufficient?

- ☑ Does the performing organization agree the task schedule is sufficient?
- ☑ Have the necessary documents to support procurement of a contracted effort been prepared (the Request for Qualifications and/or Proposal)?
- ☑ Are the Project Plan and any supporting plans documented?

Are there any recommendations that can help?

Preparing a budget for each task

To prepare the budget for each task, either you must allocate a pre-defined budget to the various tasks or you must establish the needed funds for each task (based on your task descriptions) and obtain the funds from your organization. The starting point for either approach is to estimate the effort and resources needed for each task and to convert them into a cost.

Describing each task

There are at least three parts that must be carefully defined for each task description:

- INPUTS – The information and products that must be available to the team that will perform this task.
- PROCESS – How the task should be performed
- OUTPUTS – The products of this task

These task descriptions may be organized into a Work Breakdown Structure (WBS). A WBS provides a hierarchical structure of all tasks and sub-tasks of the project, identifying the name of the task or sub-task, the allocated budget, and the team or organization with the authorization and responsibility to perform the task.

Minimum contents of a Project Plan

At a minimum, a Project Plan should include:

- Project goals and purpose
- Project task descriptions
- Project budget allocated to task
- Project reserve for contingencies
- Resources needed for each task
- Project organization chart
- Project products and deliverables
- Project schedule

Sometimes part of a schedule may be incompletely defined at this point because substantial work (work defined in one of the project's tasks) must be done to define this part of the schedule.

Supporting management plans that may be needed

Beyond the Project Plan, additional plans may be required and their preparation should be as part of the project's tasks. Among the most common such plans are:

- A Systems Engineering Management Plan (discussed in the follow section, 4.3.2)
- A Configuration Management Plan (to capture and control changes to the project's products, see section 4.8.6)
- A Risk Management Plan (to identify and mitigate major program risks, see section 4.8.4)
- A Quality Assurance Plan (to ensure the quality of the project's products)
- A Project Safety Plan (if the project involves or produces items that may be dangerous to people)
- A System Security Plan (if the system needs to be protected against external threats)

Procurement decisions

One of the most critical decisions for the project manager is to decide which activities should be done in-house by the system owner's organization and which activities should be done by another agency or by a consultant, or system integrator. In general, each task (and in some cases subtasks) should be the subject of a procurement decision. Use of some in-house resources may be mandated by agency policy. In other cases you may want to use in-house resources in order to develop a needed in-house capability, such as a software maintenance capability. On the other hand, a capable in-house resource might be reserved for other higher priority work and resources brought in for this one-time effort.

4.3.2 Systems Engineering Management Planning

OBJECTIVE:

The Systems Engineering Management Plan (SEMP) is the repository for project technical plans. The Systems Engineering Management Plan identifies what items are to be developed, delivered, integrated, installed, verified and supported. It identifies when these tasks will be done, who will do them, and how the products will be accepted and managed. It largely becomes the project control document. Finally, it defines the technical processes to be used to produce each of the project's products.

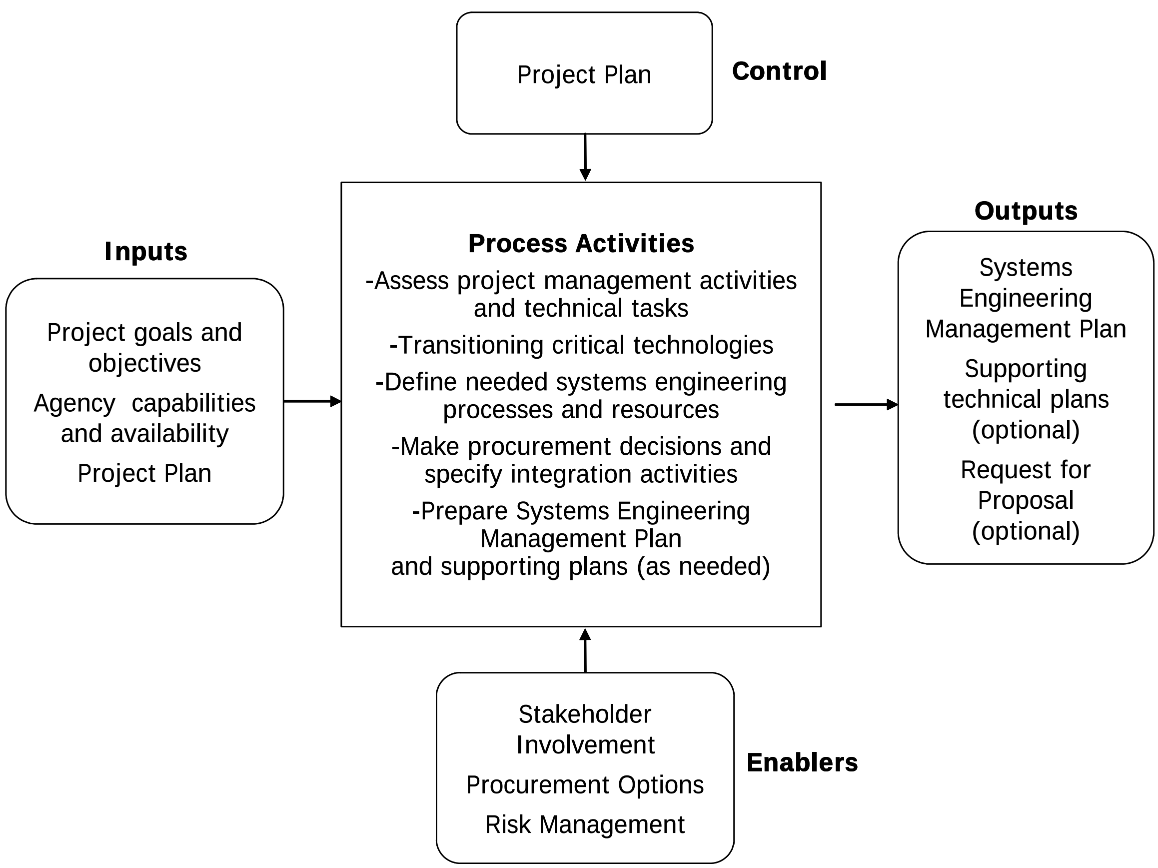
DESCRIPTION:

The SEMP is an extension of the Project Plan and focuses just on the technical tasks (the tasks covered in this Guidebook).

Preparation of the SEMP is a two-step process, first by the system owner and second by the development teams. First, the system owner develops a framework for the SEMP. This includes the organizational structure, a master schedule for the system implementation and identification of the technical tasks. For each task, the SEMP framework identifies the required outputs and, to the extent possible at this stage, the inputs and processes to be performed. The SEMP framework may define a number of other items including a candidate set of supporting plans, metrics to measure technical performance and the criteria for technical reviews. The SEMP framework also will tailor the technical processes commensurate with the scope and risk level for the project.

Second, the selected project development teams (which may be agency or contractor personnel, but are expert in the processes to be used) will take the agency's SEMP framework and supply the needed detail for the processes to be used. This will include preparing any supporting plans, for instance, a Software Development Plan or an Interface Control Plan.

CONTEXT OF PROCESS:



SYSTEMS ENGINEERING MANAGEMENT PLANNING PROCESS

Inputs:

Project goals and objectives as defined by planning, by the regional ITS architecture and by collected stakeholder needs and constraints

Agency capabilities and availability is the key input to agency make/buy decisions.

Project plan defines all project tasks, including the technical tasks further defined in the SEMP.

Control:

Project plan establishes a high level description of the project tasks.

Enablers:

Stakeholder involvement is needed to support the project's technical tasks.

Procurement options will be analyzed if any technical task is to be contracted out.

Risk management is key to developing a SEMP that will anticipate and deal with project problems.

Outputs:

Systems Engineering Management Plan defines the project's technical tasks (inputs, processes and outputs).

Supporting technical plans (optional) are prepared when necessary for a complex project.

Request for Proposal (optional) will be needed for any contracted effort.

Process Activities:

Assess project management activities and technical tasks:

Project management must first determine what project management and technical tasks are going to be required by the project. The needed tasks are driven by the organizational structure and the nature of the products to be delivered. This initial task involves analyzing the project's goals and objectives plus earlier concept exploration activities to specify needed management plans (configuration and/or risk) and actions such as resource allocation, training, and known constraints. It also calls for the technical documents (schedules and plans) to complete coverage of the engineering effort.

Transitioning Critical Technologies

Risks can come in many forms but usually involve products that have not been built before. These might include novel hardware applications (e.g., new vehicle detector technology), novel software algorithms (e.g., a new approach to adaptive signal control) or challenging performance requirements (e.g., response times, and bandwidth.). Each must be identified as a risk and the technical tasks necessary to address that risk must be included in the SEMP.

Define needed systems engineering processes and resources:

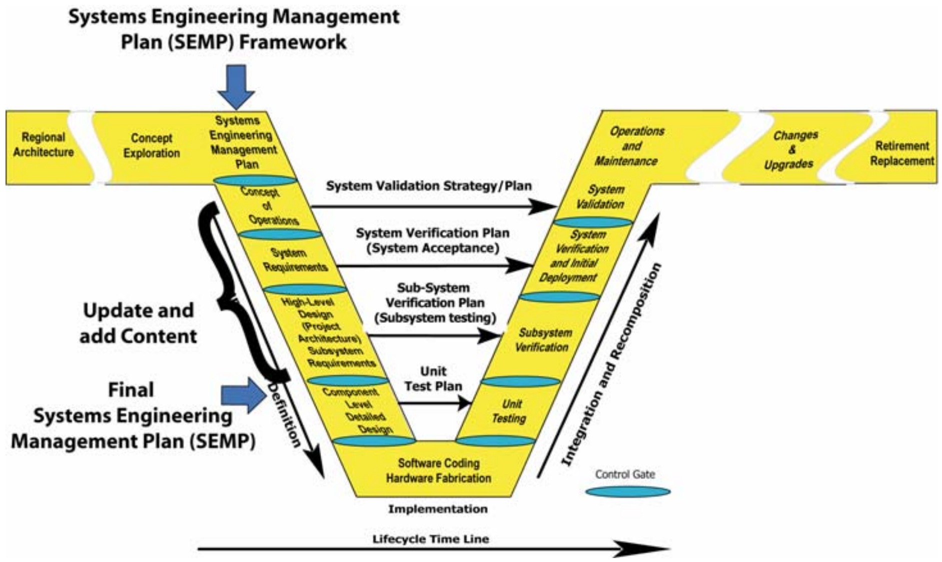
The project and engineering management will identify the systems engineering processes and resources necessary to support each identified technical task. If significant portions of the systems engineering tasks are contracted to commercial firms, they may have to be involved in detailing these processes.

Make procurement decisions and specify integration activities:

The system owner will decide, for each technical task, whether the effort can be performed in-house, consultant or system integrator. For complex engineering efforts, it is quite common to turn to consultants and system integrators. To support such procurements, the system owner will prepare the necessary contractual documents, including the Request for Proposal. The planned integration steps toward ultimate implementation ("climbing the right side of the Vee") will be specified.

Prepare Systems Engineering Management Plan and supporting plans (as needed):

In order to coordinate the technical activities between all performing organizations, the system owner, followed by the development teams, will prepare a Systems Engineering Management Plan and, if necessary, separate supporting plans such as a software development plan and other technical plans identified in the Guidebook.



Where does Systems Engineering Management Planning take place in the project timeline?

Is there a policy or standard that talk about Systems Engineering Management Planning?

FHWA Final Rule does not specifically mention general Systems Engineering Plan development practices to be followed.

The IEEE Standard for Application and Management of the Systems Engineering Process (IEEE-1220) focuses on the engineering activities necessary to guide project development. Annex B of IEEE-1220 provides a template and structure for preparing a systems engineering management plan along with an informative discussion of each section and subsection.

Which activities are critical for the system owner to do?

This is a process, like project planning, that requires careful oversight by the system owner but can, in part, be delegated to the development teams, as they are more familiar with the details of the processes to be employed. As the first stage in the completion of the SEMP, the system owner should prepare a framework that will:

- Identify the core systems engineering planning information that the developer (agency or contractor) must acknowledge to be documented during system design. Examples are work breakdown structure (schedule tasks and milestones), training, standards, and constraints.
- Identify clearly the control gates in the process where the system owner's (and other stakeholder) review and approval is required.
- In addition, the system owner must:
- Determine the resources needed for each process task and who will provide those resources (agency, consultant or system integrator).
- Select and task the performing organizations (including, as needed contractors).
- Ensure that the systems engineering analysis activities are reviewed, agreed to and documented, in the SEMP.

These tasks will vary depending on the nature of the products to be delivered, which could include such things as: designing and building custom software or custom hardware, selecting off-the-shelf hardware or software, building and evaluating prototypes, designing complex operator interfaces or a wide variety of other challenging activities.

How do I fit this step to my project? (Tailoring)

Systems engineering analysis is not one-size-fits-all. Since systems engineering analysis is there to address the technical challenges in building a system, it must be tailored to the technical challenges of the specific system.

The biggest variable affecting the scale of the systems engineering analysis is the need to develop custom software applications. Projects that only involve the purchase and installation of hardware or hardware with imbedded off-the-shelf software applications do not require nearly the depth of requirements analysis and design. Of course, there still may be serious trade studies on such issues as product selection, location site studies, or communications alternatives. The SEMP for such projects may be quite short and, for efficiency, may be combined into the Project Plan. But if custom software development is needed, then requirements definition and design become much more complex and a separate SEMP is usually the best approach.

Another factor is the degree to which the system owner is comfortable with the technologies involved. If unsure or there is a perceived risk, then added attention to the preparation of a SEMP is well advised.

The final factor to be mentioned is the degree to which the development teams have well-developed process infrastructures, for such things as tracing requirements, configuration management or software engineering. Where the agency does not have any of these process infrastructures in place, any commercial firm hired should be expected to have such well-developed process infrastructures. In such cases, the SEMP should reference these processes and only deal in detail with the unique interfaces, if any, between the existing processes and this project.

What should I track to reduce project risk and to get what is expected? (Metrics)

<u>On the technical side:</u>

- Technical performance measures (e.g. response times, and capacity.) that must be defined in the requirements and then shown to be met (simulation and modeling) by the design
- A complete end-to-end trace from user needs and the Concept of Operations to the delivered products

<u>On the project management side:</u>

- The completeness of the documents produced by each task and their correlation with the various technical reviews
- Prompt resolution and incorporation of stakeholder comments to the documents and the technical reviews
- Compliance with the systems engineering analysis processes documented in the SEMP

Are all the bases covered? (Checklist)

☑ Are all needed process steps identified, along with their process, inputs and outputs?

☑ Are all known requirements and constraints on the design (specific hardware and COTS software products) incorporated into the process steps?

☑ Are all necessary technical reviews identified and planned?

☑ For each process task, is the performing organization, and any other needed resource, identified and made available?

☑ Is the required content of each deliverable document clear to the performing organization?

☑ Is the format for delivery of custom software and its supporting documentation clear?

☑ Is the Configuration Management Plan clear on who needs to approve changes to any baseline?

☑ Has a selection committee and the selection criteria been established to support each procurement activity?

☑ Do the design, integration and verification plans support the deployment goals for the system?

☑ Are project risk areas adequately addressed by the defined tasks of the SEMP?

Are there any recommendations that can help?

An adequate level of commitment to project management is essential for ensuring the effective delivery and operation of ITS projects. Industry process standards for information technology systems point to the use of the SEMP as that engineering plan for technical control. Although, not specifically called out in federal regulation, the SEMP is considered a critical means of addressing accountability for ensuring both efficient and effective results of any systems engineering.

To the extent possible, the SEMP should plan for all disciplines (development teams) required during the project lifecycle being involved in each of the technical tasks. At a minimum, this means that some hardware and software design engineers should be involved during the very first tasks of the project, including elicitation of user needs, preparation of a Concept of Operations and in requirements analysis. Likewise, some of the systems engineers who developed the requirements should stay involved in the project through the design, production, integration, verification, and deployment tasks. This will integrate the processes and help ensure that the final system meets the original project goals.

4.3.3 Concept of Operations

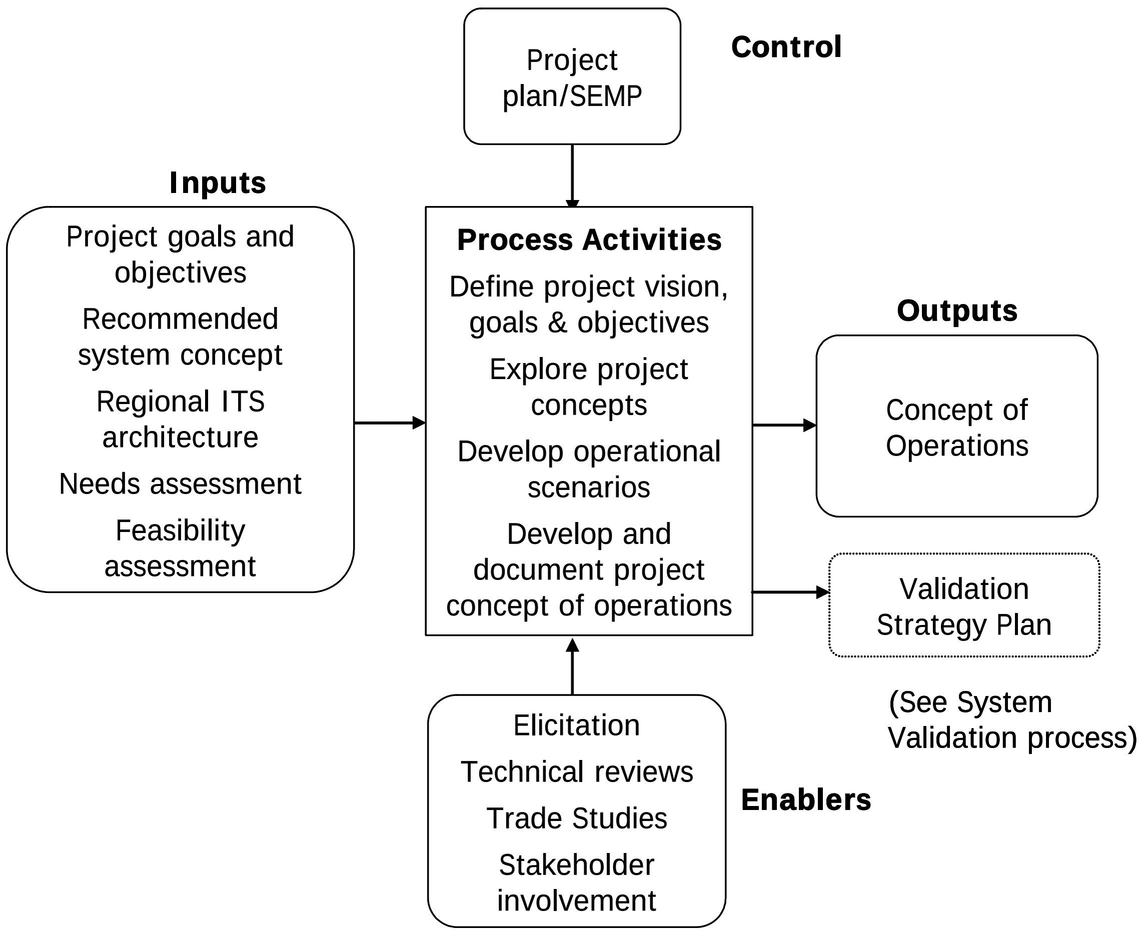
OBJECTIVE:

The Concept of Operations documents the total environment and use of the system to be developed in a non-technical and easy-to-understand manner. It presents this information from multiple viewpoints. It provides a bridge from the problem space and stakeholder needs to the system level requirements.

DESCRIPTION:

The Concept of Operations document results from a stakeholder view of the operations of the system being developed. This document will present each of the multiple views of the system corresponding to the various stakeholders. These stakeholders include operators, users, owners, developers, maintenance, and management. This document can then be easily reviewed by the stakeholders to get their agreement on the system description. It also provides the basis for user requirements.

CONTEXT OF PROCESS:



CONCEPT OF OPERATIONS PROCESS

Inputs:
Project goals and objectives determine how the system will be used.

Recommended system concept describes the concept selected for best benefit for the cost, which will be the basis for the concept of operations.

Regional ITS architecture will provide roles and responsibilities of the primary stakeholders and the systems they operate, which may suggest features for the project concept of operations.

Needs Assessment includes the list of collected needs, their sources, and documentation of the rationale for the selection of the key needs and any constraints that exist that may limit possible solutions to the needs. The development of the Concept of Operations starts with these needs and constraints.

Feasibility assessment or FSR defines and analyzes the conceptual system and, in the process, provides operational information.

Control:
The *Project Plan* describes the project and the *SEMP* describes the systems engineering effort needed for development, so they both guide what may be developed.

Enablers:
Elicitation supports continual stakeholder input and review, which is essential to developing a system meeting the needs.

Technical reviews support continuing communications with the stakeholders, which are essential to develop a concept that reflects their needs, organizations and standard operations.

Trade studies used for the concept selection support this document and may be expanded.

Stakeholder involvement is essential to be sure that the system will operate in a way that is meaningful and useful to them.

Outputs:
Concept of operations describes the operation of the system being developed from the various stakeholder viewpoints. It documents the users requirements for ultimate system operations. The users and other stakeholders can review the document and provide feedback and validation on these key going-in assumptions.

Process Activities:
Define project vision, goals, and objectives

Revisit the vision, goals and objectives identified in Concept Selection and Feasibility Assessment (Section 4.2.3). Expand and elaborate on them to capture the multiple viewpoints.

Explore project concepts

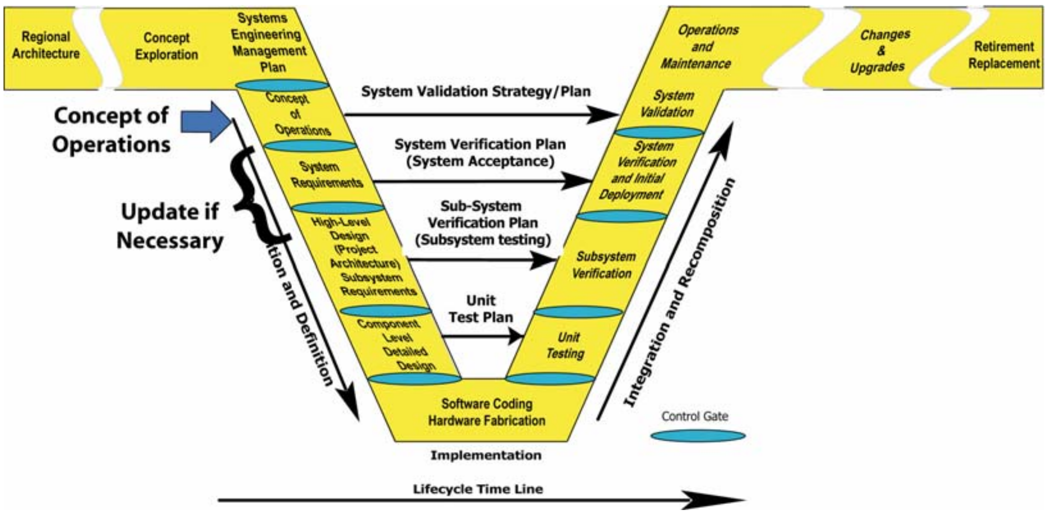
Revisit the alternative concepts identified during Concept Selection and Feasibility Assessment (Section 4.2.3). The goal is to glean just enough of a physical description of the system from the high-level system architecture to write the Concept of Operations. Perform additional trade studies as needed.

Develop operational scenarios

Operational scenarios describe how the system will be operated under various conditions. For example, incident management scenarios will include normal monitoring, the sequence of events following an incident, and response to failure (e.g., sensors or communications). These scenarios will describe the activities from the viewpoint of each of the participants. Some techniques for describing the scenarios are flow diagrams and use cases which are part of the unified modeling language used for software development.

Develop and document the concept of operations

The Concept of Operations is finally a document that records these findings and system characteristics from each of the multiple viewpoints of the various stakeholders and in a language that they each understand. This document includes such information as vision, goals and objectives, operational philosophies, operational environment, support environment, operational scenarios, operational system characteristics, system constraints and limitations, institutional issues, external interfaces and stakeholder functions, responsibilities, capabilities and interfaces.



Where does the Concept of Operations take place in the project timeline?

Is there policy or standard that talk about the Concept of Operations?

FHWA Final Rule (23 CFR 940.11) requires that participating agency roles and responsibilities be identified in the systems engineering analysis for ITS project funded with Federal money from the Highway Trust Fund, including the Mass Transit Account. It also requires that procurement options be assessed and that procedures and resources necessary for operations and management of the system be determined.

For more description of the Concept of Operations, see IEEE Standard P1362 V3.2, http://www.ieee.org and ANSI/AIAA G-043-1992 Guide for the Preparation of Operational Concept Documents, http://global.ihs.com .

Which activities are critical for the system owner to do?

- Discuss visions, goals, needs, expectations, practices and procedures, normal activities, constraints, environment and other inputs to the Concept of Operations
- Identify stakeholders
- Review the developing Concept of Operations
- Review and approve the final Concept of Operations

How do I fit these activities to my project? (Tailoring)

The level of each activity should be appropriately scaled to the size of the project. For example, a small project may have a Concept of Operations only a couple pages long. The emphasis on the concept exploration depends more on the newness of the project than on its size. For example, if the system will be automating activities that were formerly manual, or integrating formerly independent activities, it is a good idea to look at alternative ways for structuring the system. This will be useful to allow the stakeholders to envision using the new system. Whenever formerly independent activities are merged it is essential to carefully spell out the new operational responsibilities of each agency. Examples are neighboring agencies sharing responsibility for traffic management or signal control.

What should I track in this process step to reduce project risks and get what is expected? (Metrics)

On the technical side:

- Number of operational changes that the new system will require, since they introduce institutional, operational, and acceptance risks.
- Number of interfaces between formerly independent systems, since they introduce institutional, operational, and technical risks.

On the project management side:

Number of stakeholder groups who have reviewed and approved the concept of operations

Are all the bases covered? (Checklist)

- ☑ Is the Concept of Operations documented in an easily understood manner?
- ☑ Are the operations described from the viewpoints of all key stakeholders?
- ☑ Are both normal and failure operational scenarios included?
- ☑ Does the Concept of Operations cover the key information?
- ☑ Identification of stakeholders and their responsibilities
- ☑ Goals, objectives, vision

- ☑ Constraints and metrics
- ☑ External interfaces
- ☑ Operational and support environment
- ☑ Alternative concepts and rationale for the selection
- ☑ Operational scenarios
- ☑ Has the Concept of Operations been reviewed and accepted by the stakeholders?

Are there any recommendations that can help?

The Concept of Operations has applicability beyond this phase in the development. It will be used to validate the completed system since it describes how it is expected to operate for comparison and provides a theory of operation as a basis for training and user documentation for the delivered system. (add Pam comment 37)

There is a temptation at this point to make assumptions about system design. The Concept of Operations should address what is to be done, but not how it will be implemented, which will be determined later during design.

A closer look at scenarios – scenarios are an important part of the Concept of Operations. They should include, at a minimum, what is to be done, who will do it and what is communicated to whom. This could be a flowchart or text. It must be something that is easily understandable by the stakeholders. A simple way to do this is to write the scenario from the viewpoints of each of the stakeholders involved. Some other techniques that you may see used in concepts of operation are use cases, thread analysis and flow analysis.

Here is a simple example of a text scenario for a transit system from the view of the dispatcher. There will be corresponding scenarios for the driver, maintenance and the bus yard.

Scenario: Bus breakdown

Viewpoint: Dispatcher

Receive notification of breakdown from the driver.

Locate bus.

Request repairs from maintenance department.

Request replacement bus from the bus yard.

Confirm actions complete.

Notice that this scenario does not specify how these steps will be completed (e.g., means for communicating or for determining the bus location).

This scenario is short and easy to present to the stakeholders. Their feedback at this point will prevent redesign later. For example, the maintenance department may say that they always contact the yard when they are called for a breakdown, so the dispatcher does not need to do that. A manager may point out that they needs to have the actions logged. These changes are easy to make now.

Multiple viewpoints The most important purpose of the Concept of Operation is to get agreement from the stakeholders on:

- their responsibilities,
- how the system will operate,
- the environment,
- system expectations,
- processes that the system will support

This is best done by presenting the information from the viewpoint of each of the stakeholders, so that they can readily review and respond to it. Be sure that the document addresses itself at least to the operator, user, owner, developer, maintenance and management. It should answer the “five Ws and an H” that reporters are supposed to address in their writing: who, what, when, where, why, and how.

The environment in which the system will operate arguably has as much influence on system performance as does the system itself. This includes not only the physical environment, but also the political, procedural and operational environments and any other factors that either support or constrain system operation.

The following considerations circumscribe the system to be developed and should be addressed in the Concept of Operations:

- Mission objectives and rationale
- Operational philosophies
- Operational system characteristics
- System constraints and limitations
- Relevant stakeholder/developer/user organizations and policies
- External interfaces and requirements

Closer to the system are the operational and support environments. The operational environment describes under what conditions the system will be used. For example, will the operators be doing multiple tasks (such as driving) while operating the system? The support environment includes such things as maintenance, disposal, facilities, and utilities.

4.4 System Definition

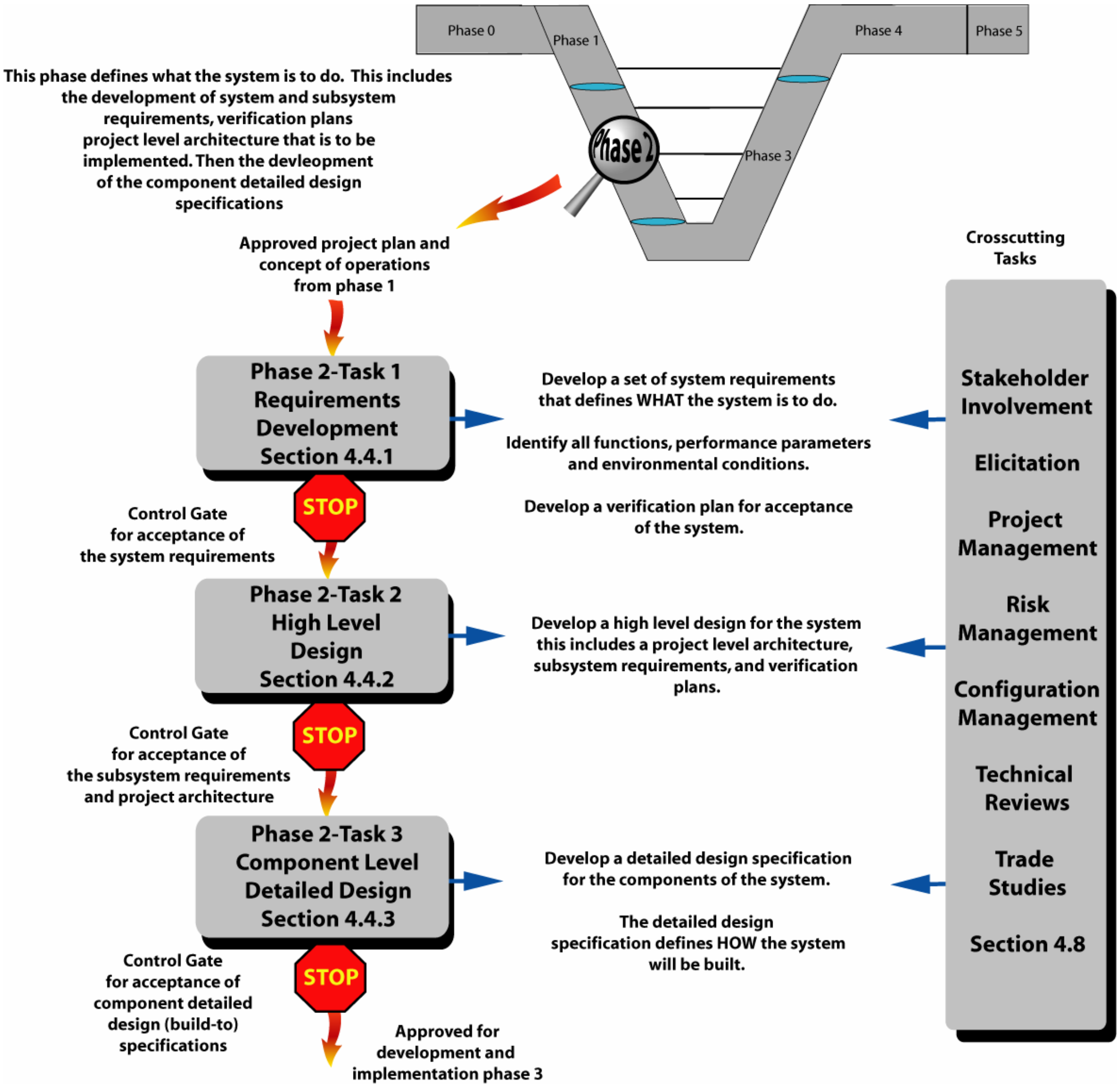


Figure 4-8 Phase 2 - System Definition Roadmap

(See comments on previous roadmap)

4.4.1 Requirements Development (System and Sub-System Level Requirements)

OBJECTIVE:

Requirements are the foundation for building Intelligent Transportation Systems (ITS). They are used to determine *WHAT* the system must do and drive the system development. Requirements are used to determine (verify) if the project team built the system correctly. The requirements development process identifies the activities needed to produce a set of complete and verifiable requirements.

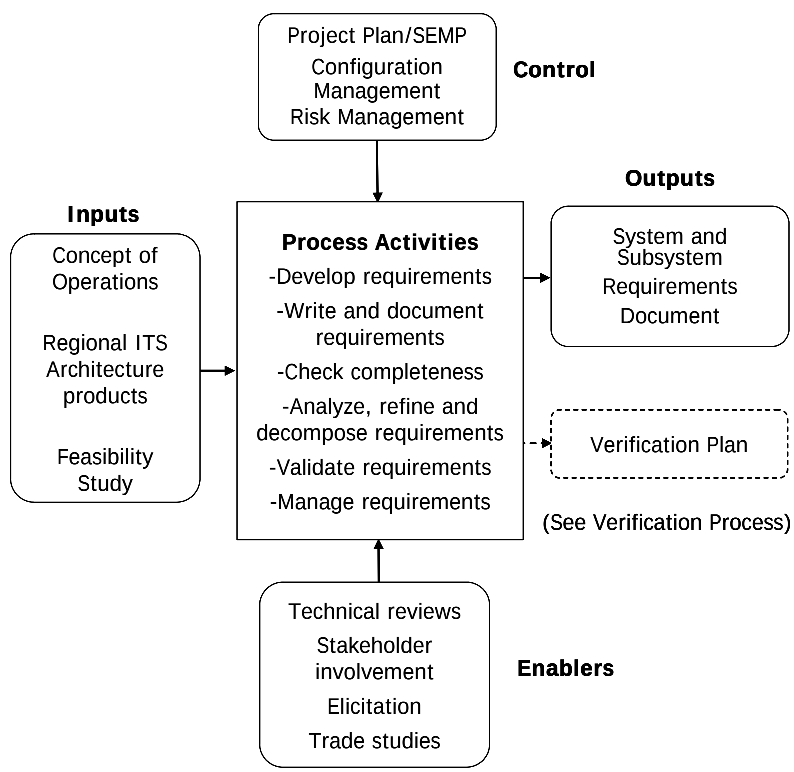
DESCRIPTION:

Requirements development is a set of activities that will produce requirements for the system and sub-systems. What is a requirement? The definition from the systems engineering standard (EIA 632) is "Something that governs what, how well, and under what conditions a product will achieve a given purpose." Simply stated, requirements define the functions, performance and environment of the system under development to a level that can be built:

- Does the system do *WHAT* it is supposed to do? - These are ***Functional*** requirements.
- How well does the system do its functions? - These are ***Performance*** requirements.
- Under what conditions (e.g. environmental, reliability, and availability.), does the system have to work and meet its performance goals? – These are ***Environmental and Non-Functional*** requirements.

There are other types of requirements that are also needed but are often overlooked. These are called enabling requirements. These define other aspects of systems development that are needed but do not show up as part of the system. For example, development, testing, support, deployment, production, training, and in some cases disposal. Primarily the *Functional*, *Performance*, *Environmental* and *Non-Functional* Requirements are contained in the System and Sub-system requirements documents. The *enabling* requirements may also be in these documents but they mainly show up in the various plans (SEMP and project plan), statements of work for contracted work, and memorandums of understandings among participating stakeholders.

CONTEXT OF PROCESS:



REQUIREMENTS DEVELOPMENT PROCESS

Inputs:

Concept of Operations documents the user needs, expectations, goals, and objectives. It contains the issues that need to be addressed by the system and describes the way the system is intended to operate from the user's perspective.

Regional ITS Architecture defines the regional framework (environment) in which this project must operate. Major external interfaces, high level functional requirements, and stakeholders are identified.

Feasibility Study produces a conceptual high-level design and requirements that can be used as a starting point for the project.

Control:

Project Plan/SEMP contain the various plans such as the review plans, configuration management plans, risk plans that control the requirements development.

Configuration management (CM) identifies the process to control changes to the requirements and manage the baseline documentation.

Risk management is used to monitor, control and mitigate high risk requirements

Enablers:

Technical reviews are used to identify defects, conflicts, missing or unnecessary requirements. Then the requirements review control gate (formal review) is used to approve the final set of requirements.

Stakeholder involvement is essential for validating the requirements (are these the correct requirements?).

Elicitation enables the discovery and understanding of the needed requirements.

A technical *trade study* is used to analyze and compare alternative requirements and their technical and cost impacts on the system.

Outputs:

System and Sub-System Requirements Documents complete, verifiable and validated. After formal review and approval (Signed off by the System Owner and/or Stakeholders) it is put under configuration control.

Verification Plan – (from the verification process) documents the plan to verify each system requirement

Processes Activities:

Develop requirements

The first step is to develop requirements from the stakeholder needs and input products. Once the requirements are documented, they are prioritized, de-conflicted and validated with the stakeholders.

Write and document requirements

The characteristics of "good" system requirements are that they should be necessary, testable, clear, concise, technology-independent, feasible, and standalone. Requirements must be documented in order to establish the base to build upon (called a baseline) and for managing changes to the requirements.

Check completeness

A complete set of requirements includes all system functions that are needed to fully satisfy the stakeholder needs, with their associated performance, environmental, and other non-functional requirements.

Analyze, refine, and decompose requirements

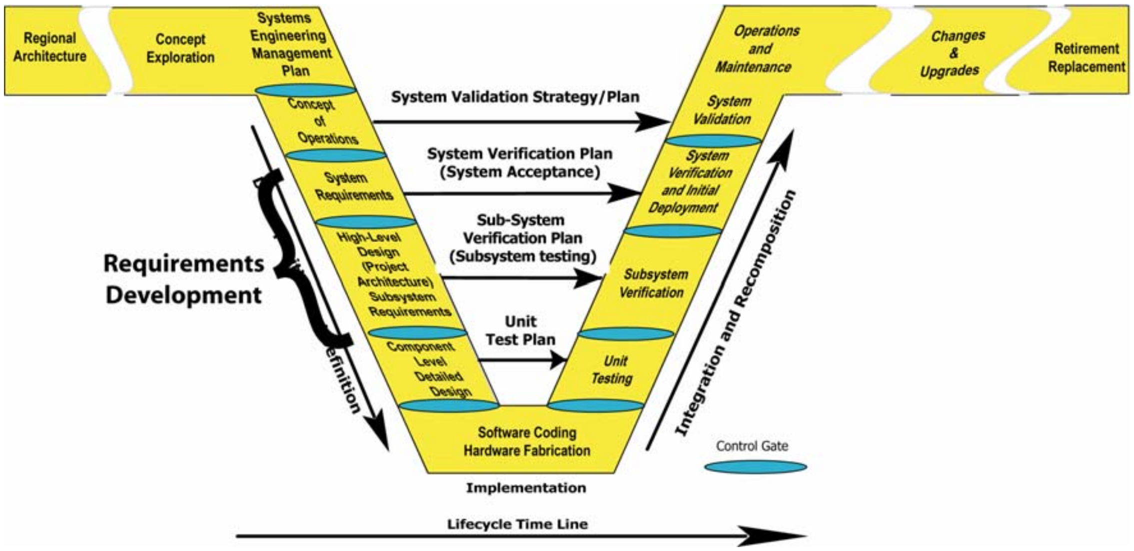
This process examines each requirement for its characteristics to see if it is a "good" requirement. Each requirement will be decomposed into a more refined set of requirements, allocated to sub-systems, and its performance requirements defined. The design is analyzed (see high level design process) for performance. New derived requirements are expected to emerge from this process. This process will continue until all requirements are defined and analyzed and the final project architecture is defined.

Validate requirements

Each requirement must be validated to ensure that these are the correct requirements. This will be done through stakeholder walkthroughs and tracing requirements to an associated need.

Manage requirements

Once the requirements have been accepted and a baseline established by the stakeholders, changes to requirements are controlled using a change management process.



Where does the Requirements Development take place in the project timeline?

Is there a policy or standard that talks about Requirements?

FHWA Final Rule (23 CFR 940.11) requires that the requirements will be developed for ITS projects funded with Federal money from the Highway Trust Fund, including the Mass Transit Account. The IEEE 1233 Guide for developing system requirements specifications provides a standard for developing requirements.

Which activities are critical for the System Owner to do?

- Assist in gathering requirements and getting the correct stakeholders involved.
- Review requirements to make sure that they are complete and address all of the needs.
- Participate in requirements walkthrough and make sure the correct requirements are being developed (validating the requirements)
- Gain stakeholder approval support for the requirements.
- Track the requirements development activities.

How do I fit these activities to my project? (Tailoring)

In this activity, there are no real shortcuts. Requirements development is a critical process for new systems. On small systems, the owner may be able to reduce the number of requirements documents by combining the system and sub-system requirements.

What should I track in this process step to reduce project risks and get what is expected? (Metrics)

On the technical side:

- Changes to requirements (high priority, cost, and risk) lead to increased cost and increased technical risk. The goal is to minimize changes to requirements after baseline.
- An incomplete set of requirements leads to increased technical risk and increased cost. The goal is to track the number of requirements that have been fully defined, analyzed and decomposed.

On the project management side:

- The number of completed requirements should match the schedule and work plan – The goal is that the completion rate of requirements should match or exceed the plan prediction.
- The growth in the number of requirements after the baseline has been established will lead to scope creep.

Are all the bases covered? (Checklist)

- ☑ Were the requirements documented?
- ☑ Was a requirements walkthrough held to validate the requirements?
- ☑ Was each requirement checked to see that it met all of the following?
  - ☑ Necessary (trace to a user need)
  - ☑ Concise (minimal)
  - ☑ Feasible (attainable)
  - ☑ Testable (measurable)
  - ☑ Technology Independent (avoid “HOW to” statements unless they are real constraints on the design of the system)
  - ☑ Unambiguous (Clear)

- ☑ Complete (function fully defined)

☑ Was a verification plan “Test case” for each requirement developed? (test, demonstration, analysis, inspection)

☑ Was each user need fully addressed by one or more requirements?

☑ Is the requirement set complete? As follows:

- ☑ functional
- ☑ performance
- ☑ enabling (training, operations and maintenance support, development, testing, production, deployment, disposal)
- ☑ data
- ☑ interface
- ☑ environmental
- ☑ Non-functional (reliability and availability).

Were attributes (quality factors) assigned to each requirement? (Priority, risk, cost, owner, date and verification method) Verification methods could include demonstration, analysis, test, and inspection).

Were the requirements reviewed and approved by the stakeholders and a baseline (reference point for future decisions) established?

During this process step, were periodic reviews performed? Were the reviews done in accordance with the review plan documented in the SEMP?

Are there any other recommendations that can help?

Requirements development activity

CAUTION

Give ample time to this activity. This is an area of high stakeholder involvement. This activity addresses risk early in the development cycle where the cost impacts are low instead of later where the cost impacts are high.

Do not approve (Baseline) the requirements too early. Give ample time to develop a set of requirements that are complete and well written. *Once developed and approved, the requirements baseline* will need to be managed using a change control process (See section 4.8.6). Changes e.g. additions, changes or deletion of requirements, after the baseline has been established normally will mean a cost and/or schedule change. (Scope creep or loss of functionality)

***Tools** will help in managing requirements on large ITS systems with hundreds of requirements tools will be essential.* There are a number of tools that can help in the development of requirements. These tools manage the tracing of requirements, handling of attributes and, in some of the high end tools, perform configuration management for requirements. For an extensive list of tools please see http://www.incose.org .

A closer look at attributes**, **baseline**, **and completeness of requirements:

***Attributes** are user defined quality factors assigned to each requirement.* Some of the more common attributes used are:

- Author (Who requested it?),
- Date (When was it requested?),
- Owner (Who is responsible for completing it?), risk
- Cost (Low, medium or high),
- Priority (How important is this requirement?).

TIP

These *attributes can help track the technical* and project performance. Attributes help in sorting and monitoring requirements. Requirements management tools have features that allow for managing these attributes along with the requirements.

***Requirements Baseline (reference point)** is the requirements document formally approved by the System Owner and stakeholders?* If so, all future development and project decisions are based on the requirements baseline. New requirements that are added, and existing ones changed or deleted, would be controlled closely using a change management process identified in the Configuration Management Plan. Once changes have been approved by the stakeholders, a new baseline would be established.

***Completeness of requirements** is to make sure that all aspects of user needs are completely defined by a set of requirements.* There is a trap in looking at functions as “stove-pipes” in isolation of other functions. This may inhibit integration of functions, and misjudge the performance of individual sub-systems such as the communications, servers and workstations. A complete definition of the needs and the analysis of requirements through decomposition is a key activity to address completeness of the requirements.

4.4.2 High Level Design (Project Level Architecture)

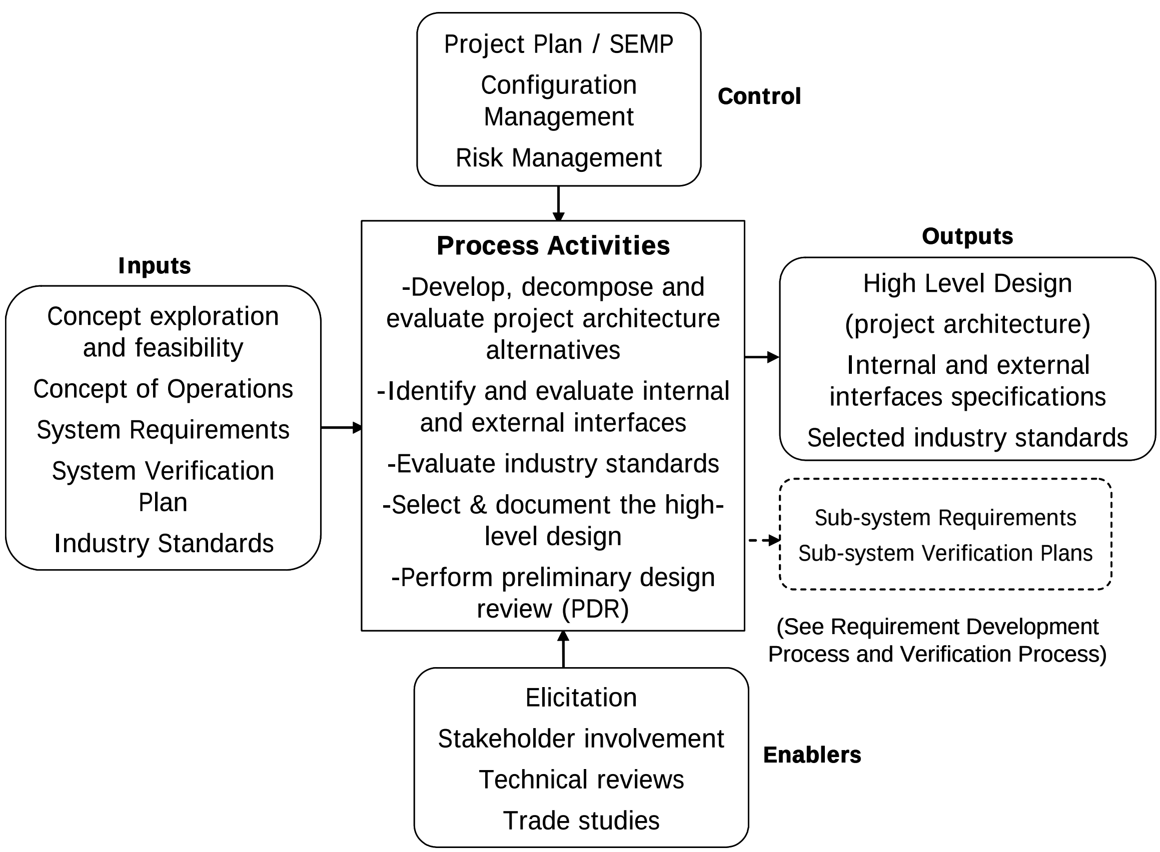
OBJECTIVE:

The high level design defines the project level architecture of the system. This architecture defines the sub-systems to be built, internal and external interfaces to be developed and interface standards identified. The high level design is where the sub-system requirements are developed. The high level design also identifies the major candidate off-the-shelf products that might be used in the system.

DESCRIPTION:

High level design is the transitional step between WHAT (requirements for sub-systems) the system does and HOW (architecture and interfaces) the system will be implemented to meet the system requirements. This process includes the decomposition of system requirements into alternative project architectures and then the evaluation of these project architectures for optimum performance, functionality, cost and other issues (technical and non-technical). Stakeholder involvement is critical for this activity. In this step, internal and external interfaces are identified along with the needed industry standards. These interfaces are then managed throughout the development process. Functional and physical decomposition are the key activities that will be used. Functional decomposition is breaking a function down into its smallest parts. For example, to decompose the function ramp metering, it might include a number of sub-functions e.g. detection, meter rate control, main line metering, ramp queuing, time of day, and communications, or other items. Physical decomposition defines the physical elements needed to carry out the function. For example, the ramp metering decomposition might include loops or video detection, controller clock, fiber or twisted pair for communications, 2070 or 170 controllers, host computers, cabinets and conduits. And finally allocating these sub-functions to the physical elements of the system will form the complete project architecture. This step also defines the integration and verification activities needed when the system elements are developed.

CONTEXT OF PROCESS:



HIGH LEVEL DESIGN PROCESS

Inputs:

System Requirements are used as the primary source for the project level architecture.

Concept of Operations provides user requirements and context to the sub-systems requirements.

System Verification Plan will provide context information for sub-system verification (what the sub-system needs to do to meet the system verification). This augments the system level requirements.

Control:

Project Plan / Systems Engineering Management Plan (SEMP) defines the process for developing the design.

Configuration Management Plan defines the process for managing changes to requirements.

*Risk managemen*t monitors, controls and mitigates high risk factors of the High Level Design, architectures, requirements, and technology.

Enablers:

Elicitation supports this process, which is essential to developing a system meeting stakeholder needs.

Technical reviews support continuing communications with the stakeholders, which are essential to develop a concept that reflects their needs, organizations, and standard operations.

Trade studies are used to analyze design alternatives and to select among them.

Stakeholder involvement is needed to validate the sub-system requirements and architecture.

Outputs:

High Level Design (project architecture) is documented, controlled, moving forward into detailed design.

Internal and external interface specifications that will need to be managed.

Selected industry standards that are recommended for the High Level Design.

Sub-System Requirements and *Sub-System Verification Plans* from the requirements/verification process

Process Activities:

Develop, decompose, and evaluate project architecture/High Level Design (HLD) alternatives

Systems engineers will first evaluate several candidate architectures/HLD's that appear to meet the requirements. Using analytical tools and methods, each alternative is decomposed into simple functions that are then allocated to sub-systems then evaluated to see if this HLD meets the system requirements (functionality and performance). This process repeats until each HLD is complete.

Identify and evaluate internal and external interfaces

Interfaces should be identified as early as possible and then managed throughout the development process. Interfaces will define the boundaries of the system (external from requirements) and sub-systems (internal from HLD) and will be natural points for integration.

Evaluate industry standards

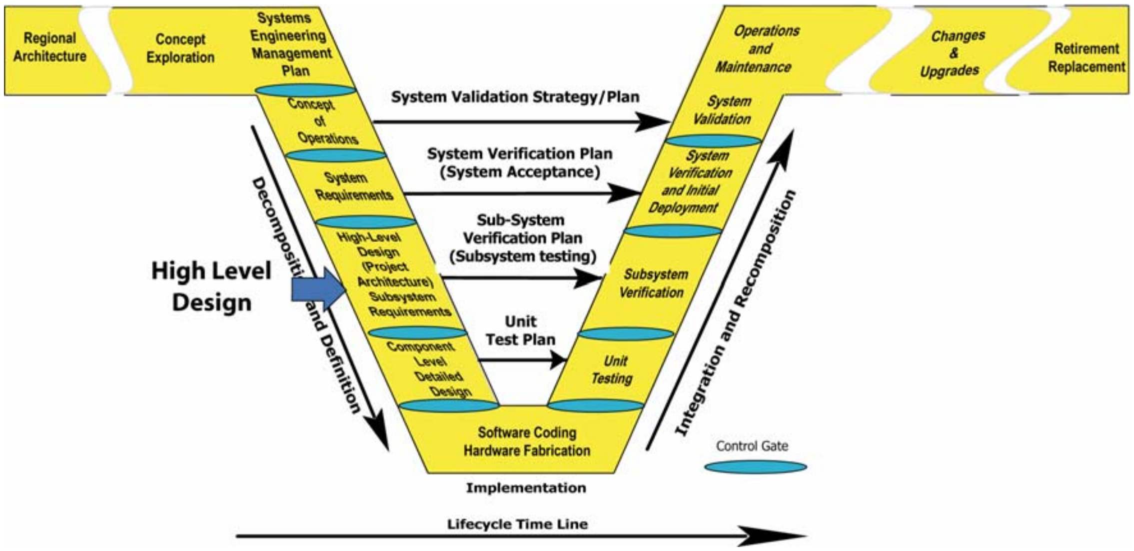
Use industry standards wherever possible. ITS systems will evolve over time and novel interfaces will be much more difficult to manage and change. Standard interfaces will tend to be more flexible and, since it is a standard, it will be easier to find products that will interface if needed.

Select and document the High Level Design

Trade studies are used for evaluating (contrast and rank) candidate architectures, if there is a clear HLD that "wins" over the other candidates it should then be presented to the system owner and stakeholders for their concurrence. Most likely two or three candidate HLD's will meet the requirements but have strengths and weaknesses in different areas. Recommended alternatives are presented to the system owner

Perform the preliminary design review

This consists of a review of the draft High Level Design document and of a design review presented to the system owner and stakeholders. The team will revise the document based on stakeholder comments and submit the final High Level Design document. Since this is the first time that sub-systems are defined, the team will develop sub-system test plans and will update the SEMP as necessary.



Where does the High Level Design take place in the project timeline?

Is there a policy or standard that talks about High Level Design?

FHWA Final Rule (23 CFR 940.11) requires that the requirements will be developed for ITS projects funded with Federal money from the Highway Trust Fund, including the Mass Transit Account. It also requires the analysis of alternative system configurations to meet requirements.

The IEEE 1233 Guide for developing system requirements specifications provides a standard for developing requirements.

Which activities are critical for the system owner to do?

- Negotiate interface agreements if the system has interfaces to other legacy systems.
- Review High Level Design alternatives.
- Participate in and review the alternative selection process, especially in determining the relative importance of various selection criteria.
- Participate in the high level design review and insure the right stakeholders are in attendance.
- Review and approve the High Level Design document.

How do I fit these activities to my project? (Tailoring)

The level of each activity should be appropriately scaled to the size and budget of the project. For example, a small project may have an analysis of alternatives only a page or two long, and based on qualitative comparisons. Constraining the number of sub-systems will also reduce the effort here and in the subsequent steps, such as integration and verification.

What should I track in this process step to reduce project risks and get what is expected? (Metrics)

On the technical side:

- The tradeoffs of functionality, performance and technology for alternative High Level Designs.
- The interfaces of the system and especially the unique and non-standard interfaces.
- Trend in design toward unproven technologies or equipment increases risks.
- Trend toward a design requiring higher development or O&M costs increases risks.

On the project management side:

- Levels of decomposition will drive the integration and verification effort adding integration costs and schedule time.
- Interfaces to external systems will require agreements that need to be developed and managed, this tends to increase schedule due to institutional issues of approvals and commitments.
- Number of identified alternatives that have been fully analyzed can be a risk of increased cost.
- Number of completed sub-system designs.

Are all the bases covered? (Checklist)

☑ Were alternative project architectures/HLD's considered?

☑ Is there documented rationale for the selected project architecture/HLD?

- ☑ Are all interfaces identified and documented?
- ☑ Have industry standards been identified for the HLD?
- ☑ Is the design clearly documented?
- ☑ Is the High Level Design traceable to the system requirements?
- ☑ Do any of the requirements need to be changed based on the High Level Design development effort?
- ☑ Have the integration, verification and deployment plans been updated in SEMP?

Are there any other recommendations that can help?

Tools and techniques are available to support high level design. These tools include functional decomposition tools, modeling tools, and management tools for tracking changes to the high level design.

As a general rule, do not specify any part of the design unless that design decision has been justified during the alternatives studies. Sometimes there is a tendency to overly detail the design at the top-level. For instance, is it necessary to identify the operating system at this time? This may unduly constrain the implementation and lead to higher development costs.

A closer look at high level design alternative project architectures, and architectural views.

Sub-systems defined in the High Level Design

The High Level Design process must define the division of the system into sub-systems. Sub-systems, and the way they relate to each other, become one element of the system architecture. Sub-systems may be needed for a variety of reasons, including:

Parts of the system that are to be developed, or procured, separately - For instance, if the system includes a wide area network to connect multiple sites, that WAN may be a sub-system with a common interface (say the input to a router) to other sub-systems.

Parts of a system that are to be deployed to different locations, or in different configurations to multiple locations

Division of a complex system into simpler parts each of which has a unifying, and somewhat independent set of functions, such as, separating display functions and components from database functions and components.

As part of defining the sub-systems, requirements must be allocated and traced into each sub-system. Sometimes it will be necessary to further decompose a requirement so that it is clear what part of the requirement is performed by an individual sub-system.

Hardware defined in the High Level Design

Hardware definition is somewhat synonymous with a physical architecture description (although there is also a software aspect to the architecture that will be covered next). Each of the architecture components (which may or may not be considered sub-systems) must be defined in terms of its hardware. The definition may be generic (e.g. a workstation, a server, or traffic signal controller) or may be specific (by manufacturer and model number) depending on the results of the alternatives studies previously done.

Software defined in the High Level Design

Usually, each sub-system would have a separately identified software component. A sub-system may have several if it contains multiple processors. The software component should be defined both in terms of its custom developed parts (the application) as well as its off-the-shelf parts, such as the operating system, database software, or communications software. Here again, these software components should be defined generically, unless an alternative study has determined that a specific product is necessary. It is for the custom designed software application that tracing of functional and performance requirements are most important.

Other aspects of the high level software design may be dependent on the design methodology used. For instance, if object oriented methods are to be used, the high level design would identify major objects of the system.

Operator interface defined in the High Level Design

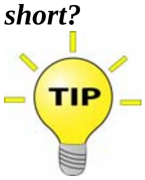
The details of the operator interface design are a critical part of the requirements of the system. It is also a part of the design that requires extensive input from the eventual users and operators of the system. Of course, if the operator interface is just an on/off switch, that is not much of a design problem. But here we are talking about a workstation interface that can be surprisingly complex. The operator interface design must describe, in detail, everything that is displayed to the operator and all actions the operator can take, via the workstation. If the display contains a map then all the contents of that map (e.g. roads, icons for loop detectors, signals, or message signs) must be defined both in terms of what it looks like as well as when it is displayed. For instance, maps are generally divided into layers of similar information and each layer can be turned on or off by the operator. Similarly, all actions by the operator to enter data or to cause things to happen (like displaying a message on a sign) must be defined. Both display and entry should be designed in ways to limit operator error, from looking at the wrong data to entering a wrong value or command.

The project management and systems engineering team for the project supports you in deciding the appropriate level of engineering and operational talent to be applied to the operator interface design. It is not a task to be left to the software programmers.

Alternative project level architectures

Based on a complete familiarity with all previous work including the concept exploration, user needs, the concept of operations, the project plan and SEMP and, most importantly, the system requirements, it is now time to start looking at the project architecture (HLD) of the planned system. Although the system requirements should be design independent, there is usually, by this time in the project lifecycle, some expectation of a functional and physical architecture that was brought forward from the concept exploration phase. The alternatives may be complete for the entire system or perhaps alternatives for just a part of the system. It is not uncommon that the various alternatives can be combined into a very large number of different configurations. Alternatives should be defined before the allocation is done, since there may be alternative allocations that should be considered. For example, loop data processing may be done at the roadside or centrally. Trade studies are used to evaluate the alternatives relative to the requirements and determine which are compliant. They will then compare the compliant alternatives in terms of cost, performance and the goals and objectives.

What to do with project architectures that fall short?



Even project architectures that fall short of meeting all requirements may provide useful information. Sometimes an otherwise promising HLD may fall short of some of the requirements, especially ambitious performance requirements. If such an HLD has some useful features, for example, lower cost, and ease of implementation, and if it is operationally feasible to ease the requirements of a project to make such a design acceptable, then it may be carried forward as an alternate solution. Certainly the degree to which such a design does not meet user needs should be an important factor. Alternative fully-compliant designs should be documented for future reference. In fact, the entire evaluation process, including the alternatives considered and not considered and the rationale for the selection and rejection, should be documented so stakeholders can review them.

There are many views that are very useful and should be used appropriately. The following are examples of different views that can be used. In the description at the beginning of this section, we focused on two views in the example, the functional view and the physical view. These are the most common ways that systems are described because they are easy to understand.

- Operational views (behavioral, dynamic)
- Information views (data, data flow)
- Network views (distributed, centralized)
- Activity view (functional)
- Physical view (hardware, software)

Operational view (behavioral, dynamic), describes how the system will react when it is stimulated. This is a dynamic modeling of the system that is important when real time operation needs to be carefully analyzed.

Information views (*data, data flow),* is used in data intensive systems where the data needs to be modeled in order to determine how the optimum system architecture will handle the information. Some examples are how much communications bandwidth will be needed and how the data is to be stored and accessed.

Network views (distributed, centralized) is used when analyzing the interactions between various system elements on complex networks. For example, this aids in understanding the addressing schema, and in analyzing protocol efficiencies for the network.

Activity view (functional), are the functions that are to be carried out by the system. (For examples, see description at the beginning of this section)

Physical view is the equipment that is used in the system. (For examples, see description at the beginning of this section)

4.4.3 Component Level Detailed Design

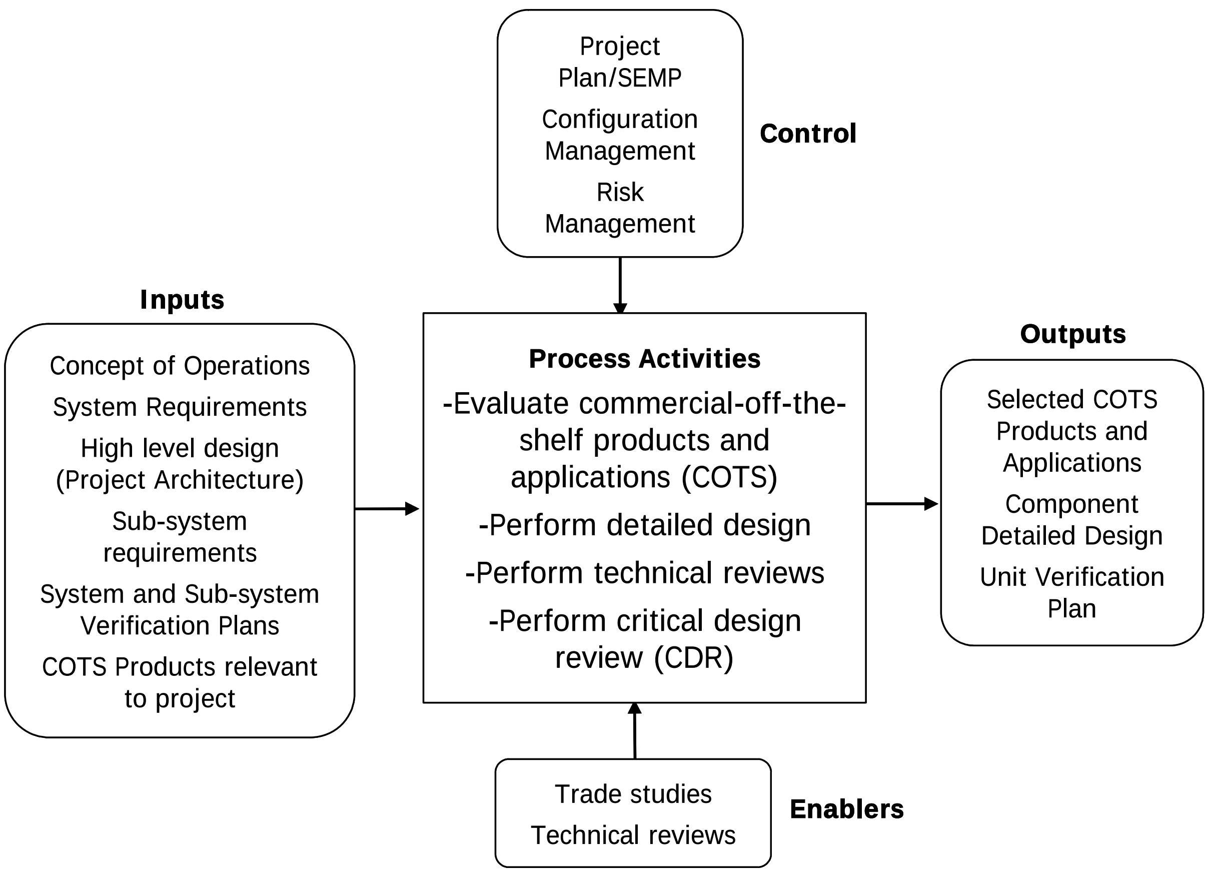
OBJECTIVE:

Component Level Detailed Design is the build-to design of the hardware, software and the selection of commercial-off-the-shelf (COTS) products. For software development, this is the step where the software documentation is being prepared for coding. For hardware, it is the step where logic schematics, chip layout, and artwork are being prepared for fabrication. If COTS equipment is being used, this step is where alternative candidate products are evaluated and selection is made.

DESCRIPTION:

Component design for software, hardware, communications, and databases describes HOW the components will be developed to meet the required functions of the system in great detail. For computer programs, this will describe the software in enough detail so that the software coding team can write the individual software modules. For hardware, this step will describe the hardware elements in enough detail to be fabricated or purchased. This level of detail is best performed by the development team who writes the software code, designs the hardware and communications, then manages the design and development process starting in this phase to the end of the development of the software and hardware. Systems engineering supports this activity by monitoring and reviewing the detailed design process and clarifies the requirements when needed. Systems engineering is involved in the periodic technical reviews during the component design process. At the completion of this step, the system owner and stakeholders will have a Critical Design Review to review and approve the "build-to" design.

CONTEXT OF PROCESS:



COMPONENT LEVEL DETAILED DESIGN PROCESS

Inputs:

Concept of Operations documents the users' needs and expectations, and provides a description of the way the system is intended to work.

System Requirements provide the designer with the overall requirements of the system and each of the system requirements should be traceable to a sub-system element.

High Level Design (Project Architecture) identifies interfaces and sub-system performance requirements.

Sub-System Requirements that each designed component should trace to.

System and Sub-System Verification Plans provide added information on how the system and sub-system is to be verified. This will assist the designer to design the components and develop the component verification procedures.

COTS products relevant to the project that will be candidates for evaluation and selection for the project.

Control:

Project Plan/ Systems Engineering Management Plan (SEMP) defines the plan for how the detailed design work will be carried out. Progress of the design work activities should be monitored against this plan.

Configuration management (CM) process should have been defined by the development team and approved by the system owner. At this step Developmental Configuration Management is used. The Developmental CM must fit into the systems owner CM plan.

Risk management is used to monitor, control and mitigate design risks, e.g. technology, and/or constraints.

Enablers:

Trade studies are used to analyze and compare alternative COTS products, detailed design alternatives and their associated impacts on the system.

Technical reviews are used to identify defects, conflicts, and missing detailed design requirements to ensure that the component design is addressing all of the sub-system requirements and is fit for the intended purpose.

Outputs:

Selected COTS products and applications are the results of the evaluation of COTS products. Ideally this is done as late as possible in the timeline to provide the latest technologies at the best price. Sometimes, however, this may have to be done earlier because of legacy systems or internal standards.

Approved Component Level Detailed Design is now ready to move forward to implementation.

Unit Verification Plan is used to verify that the components work as designed.

Process Activities:

Evaluate commercial off-the-shelf products and applications (COTS)

The stakeholders must be involved in the review of any gaps between the requirements and the COTS product specification. If there is a gap then the stakeholders should decide to use the COTS product with a deviation from the requirements, modify the product, or develop a custom application or product.

Detailed Design

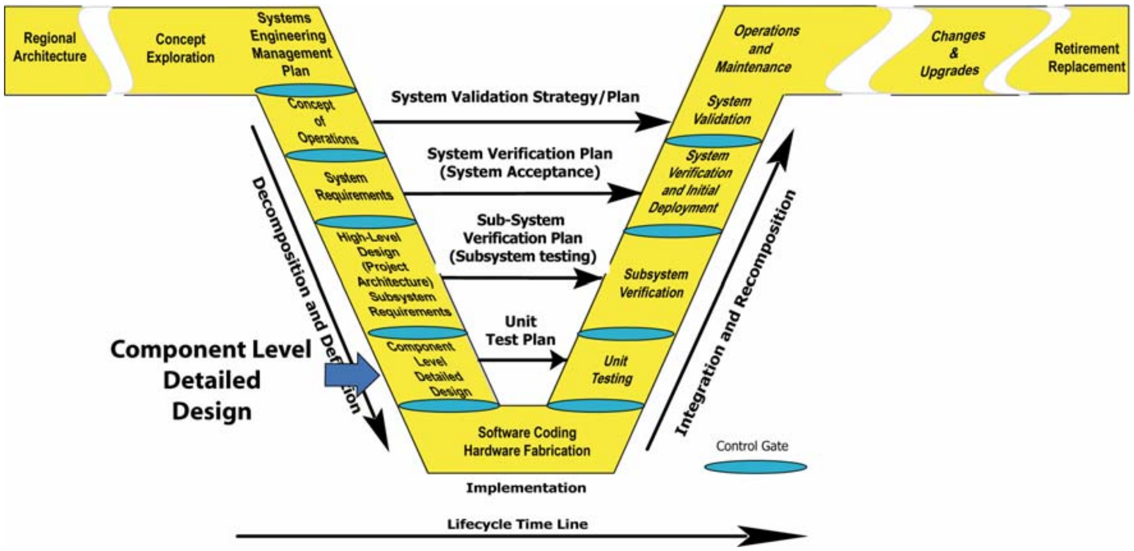
This process is performed by the development team, who will be generating the application software and integrating the hardware, databases, and communications with these applications. The development team will use a variety of techniques and tools based on the development team's approach to development, such as coding languages, methodologies, and design tools.

Perform technical reviews (performed in accordance with the SEMP)

For design status, evaluation of the candidate solutions or COTS products, technical reviews should be held on a periodic basis to review the progress of the design or selection of COTS products.

Perform critical design review (CDR) (performed in accordance with the SEMP)

At the completion of the detailed design stage, a final "build to" review is held with the Stakeholders. The purpose is for the development team to get final approval of the design prior to starting the implementation of the solution. Component design through software development to unit test is the domain of the software, hardware, and database specialist of the development team. The systems engineer needs to be able to translate user requirements to the language of these disciplines. For example, if software engineering is using UML methodology, the systems engineer needs to interface between the user needs, systems requirements and the software engineer to ensure that the design accurately reflects the intended purpose. This applies to database, hardware analog, digital logic and communications designers.



Where does the Component Level Detailed Design take place in the project timeline?

Is there a policy or standard that talks about Component Level Detailed Design?

FHWA Final Rule does not specifically mention general component level detailed design practices to be followed. For software, IEEE/ISO 12207 Software Lifecycle process provides specific process guidance.

Which activities are critical for the system owner to do?

- Participate in the technical reviews.
- Participate in the evaluation of commercial-off-the-self (COTS) products
- Participate in the critical design review.
- Approve the detailed design when completed and gain stakeholder support for the design.

How do I fit these activities to my project? (Tailoring)

This activity is driven by the amount of custom development that is needed for the project. The more custom development there is the more effort there will be at this step. For small systems that contain nearly all COTS products, the primary activity is the evaluation of these products.

What should I track in this process step to reduce project risks and get what is expected? (Metrics)

On the technical side:

- Changes to requirements (High Priority, Cost, and Risk) due to detailed design activities. Changes lead to increased cost and increased technical risk. The goal is to minimize changes to requirements.
- Incomplete design leads to increased technical risk and increased cost. The goal is to review and track the number of requirements that have been completely designed.

On the project management side:

- Number of completed designed components per schedule and development plan. The goal is that completion rate of designed components should match or exceed the plan prediction.

Are all the bases covered? (Checklist)

- ☑ Did each component have a technical review?
- ☑ Did each component design trace to a sub-system requirement?
- ☑ Were all sub-system requirements satisfied by the component design activity?
- ☑ Was a verification plan for each component defined?
- ☑ Was each component design checked for performance?
- ☑ Was the component design documentation complete, up to date, and accurate?
- ☑ Was a critical design review conducted?
- ☑ Was an alternatives analysis done on the COTS products used in the system?
- ☑ Have all system and sub-system requirements been updated at the time of the critical design review?

Are there any other recommendations that can help?

It is recommended that the development team who will be doing the software development also perform this component design activity. This continuity between the component design and development is critical.

Be sure that the development team has documented processes for developing software and hardware, and that they can share this information with you. Some development teams will be reluctant to share this information for fear of revealing this information to their competition. If so, it may require that you have a non-disclosure agreement with them. But it is important to review the development processes and have it as part of the Systems Engineering Management Plan.

Component design tools are essential for component level detailed design and there are many on the market. Each development team will have their favorite set of tools. These tools will be driven by the vendor of the tool, and the process that the development follows. This is especially true for software development.

If this is a custom development, request all tools at the completion of development. As the system owner you will want to maintain, upgrade, or change these products with or without the current development team. So the tools that are used in the component design activity need to be carefully documented. If you as the system owner had paid for these tools they need to be transferred to you with all modifications, upgrades and instructions on how they were used during the design activity. That way you can replicate this environment for future modifications.

***A closer look at software component design and development* -** *Software design is unique relative to other disciplines such as hardware, or mechanical detailed designs, in that at the component level it is tightly linked to the actual coding and implementation phase and there is a higher degree of interaction between the two phases of work.* The software process parallels the systems engineering process to a high degree as illustrated below. The team's development process should address each of the steps below. During the software design, the developer will use the system level documentation, such as the system concept of operations and system and sub-system level requirements, and revisit these from a software point of view. This is an important process if the software development team has not been involved with the system level concept of operations, or system level or sub-system requirements. In the software development environment, prototyping and spiral development methods are important tools for defining requirements. For example, prototyping graphical user interfaces for workstation will allow the stakeholder to discover features and functions that they will like or dislike before software coding. This, I KNOW IT WHEN I SEE IT (IKIWISI), is a powerful tool for software developers (see illustration below).

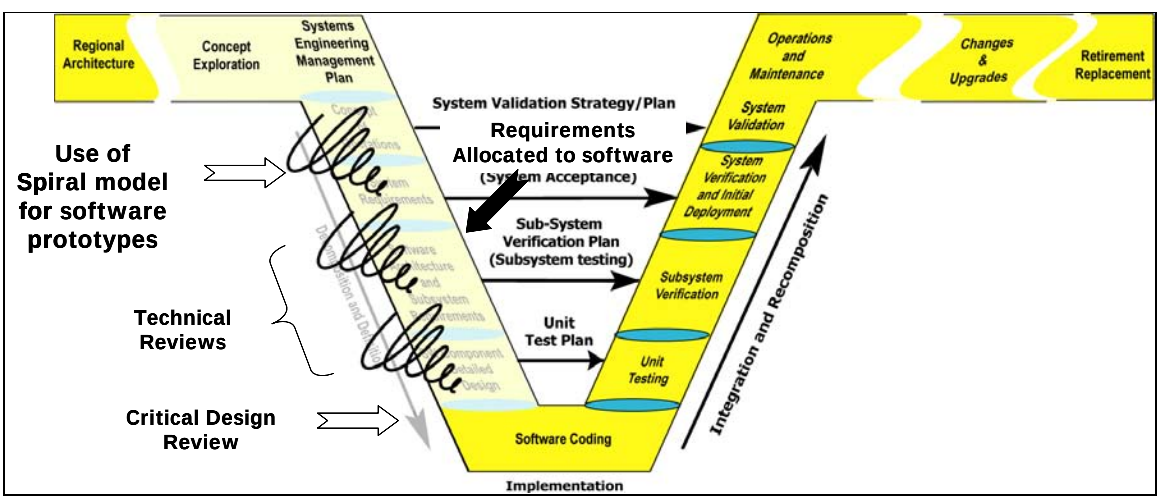


Figure 4-9 Spiral Software Development in Context with the Vee

4.5 System Development and Implementation

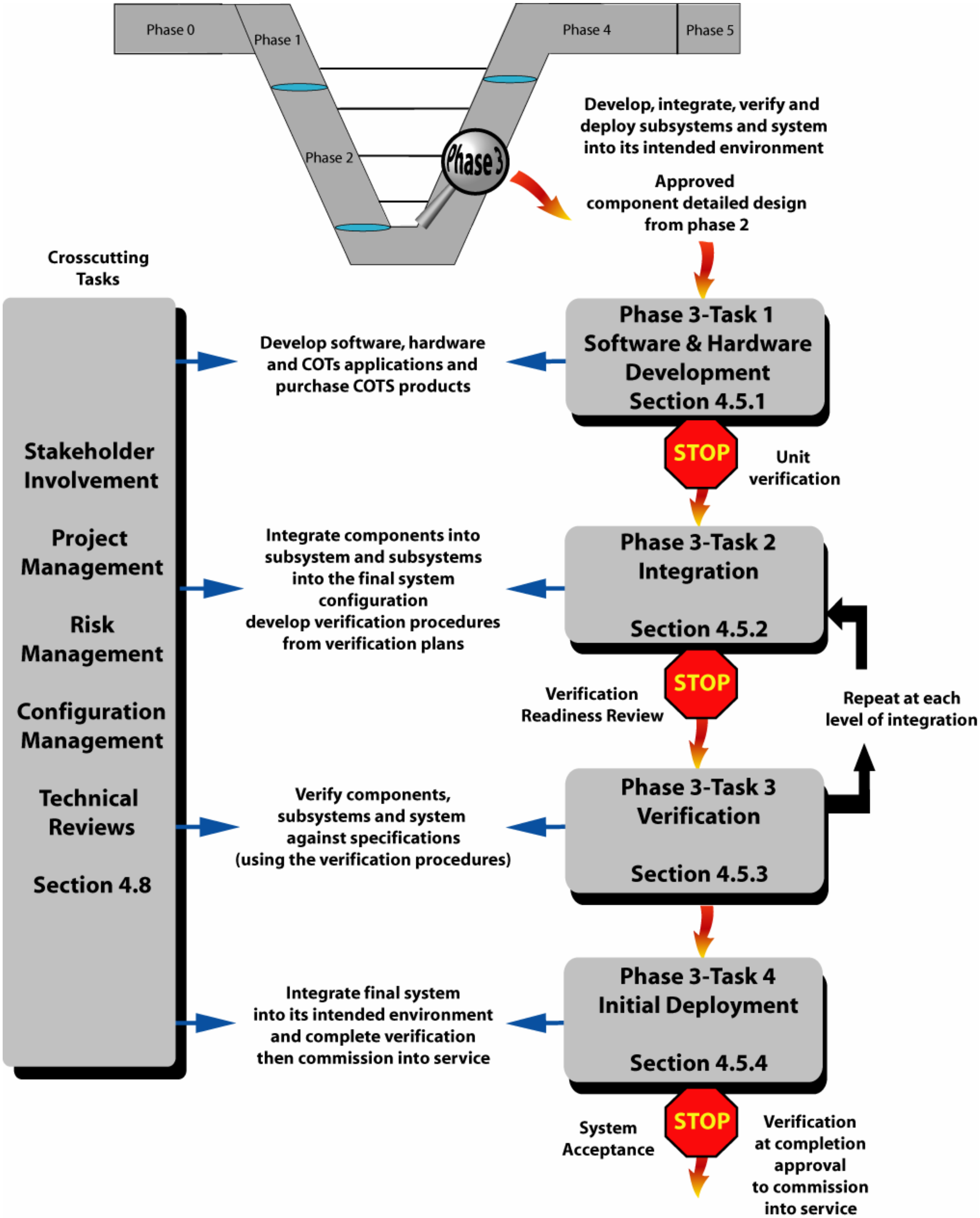


Figure 4-10 Phase 3 - System Development and Implementation Roadmap

4.5.1 Hardware/Software Development

OBJECTIVE

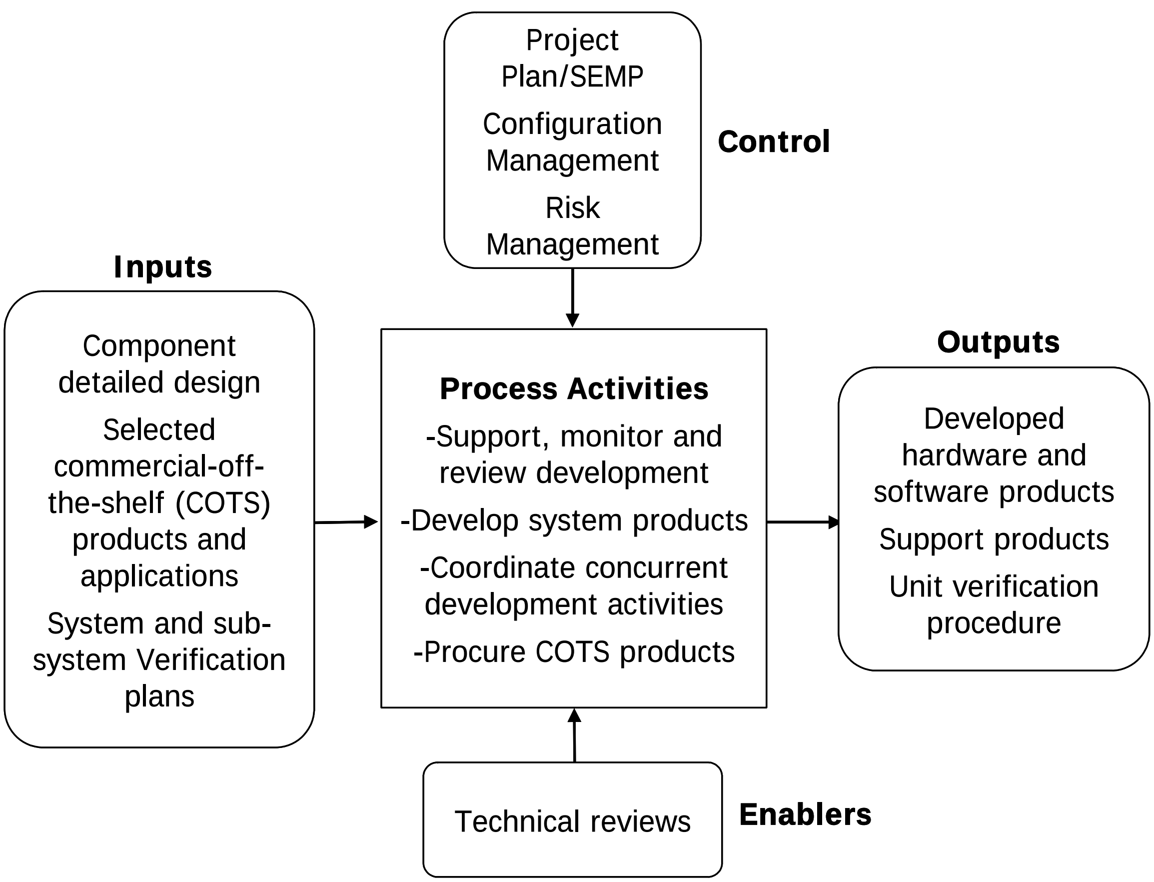
This step in the process develops (builds or constructs) the hardware and software for the system that matches the requirements and component level detailed design documentation. This step is primarily the responsibility of the development team who is fabricating the hardware and writing the software programs. The systems engineering activities are to support and review this development effort by or on behalf of the system owner. If multiple developments for the same system are underway, the systems engineering activity includes the monitoring and coordination of these developments to ensure that these projects integrate together with a minimum of effort.

DESCRIPTION:

The systems engineering activity during the development phase is monitoring and coordination of the development. The implementation is primarily the responsibility of the implementation team, whether that is done in-house or by a development contractor. Monitoring is done by a preplanned series of reviews that have been coordinated with the development team and systems engineering activity of the agency e.g. a consultant or system manager. It is essential to review the technical progress and provide technical guidance on the implementation of requirements.

These reviews provide early warning that requirements will not be met or changes, deviations or waivers will be sought. Also, these reviews will be needed when coordinating among concurrent developments for the same project depending on the development strategy.

CONTEXT OF PROCESS



HARDWARE/SOFTWARE DEVELOPMENT PROCESS

Inputs

Component Level Detailed Design is the "build-to" documentation. The coding and fabrication team develop their products based on this documentation.

Commercial-off-the-shelf (COTS) products are procured for the project. The intent is to wait until the last possible opportunity to procure technology to get the latest and most cost effective products.

System and Sub-System Verification Plans are used to assist the development team to fully understand the design and requirement that they are building to.

Control

Project Plan/Systems Engineering Management Plan (SEMP) will have the software/hardware development plan that will be used as a roadmap to carry out the software and hardware development.

Configuration Management Plan will be used to identify the needed products from the development and to manage changes during this step.

Risk management identifies, monitors, controls hardware/software development risks.

Enablers

Technical reviews are used for monitoring the project management and technical progress of the development. When multiple concurrent developments are being performed, the technical reviews can be used as coordination meetings to keep projects synchronized with each other.

Outputs

Developed hardware and software are the units or products that have been developed for the intended system. These are units of software and hardware that are now ready for integration into larger more complex functions of the target system.

Support products such as user training materials, and maintenance manuals, development and other support tools.

Unit Verification Procedures are the step-by-step instructions on how to verify that the units match the design.

Process Activities:

Support, monitor and review development

During the development phase, technical reviews should be held according to the technical review plan developed by the development team. These reviews assess the progress and technical correctness of the implementation of the design. Technical reviews give the stakeholders visibility into the development process, and build confidence in the development outcome.

Develop system products

This is where the actual software code is developed, COTS applications and hardware is fabricated for the system. In addition to these, support products are developed such as users manuals, training products, maintenance manuals are started. As integration and verification proceeds these products are updated as needed and final delivery should follow the delivery of sub-systems and the final system.

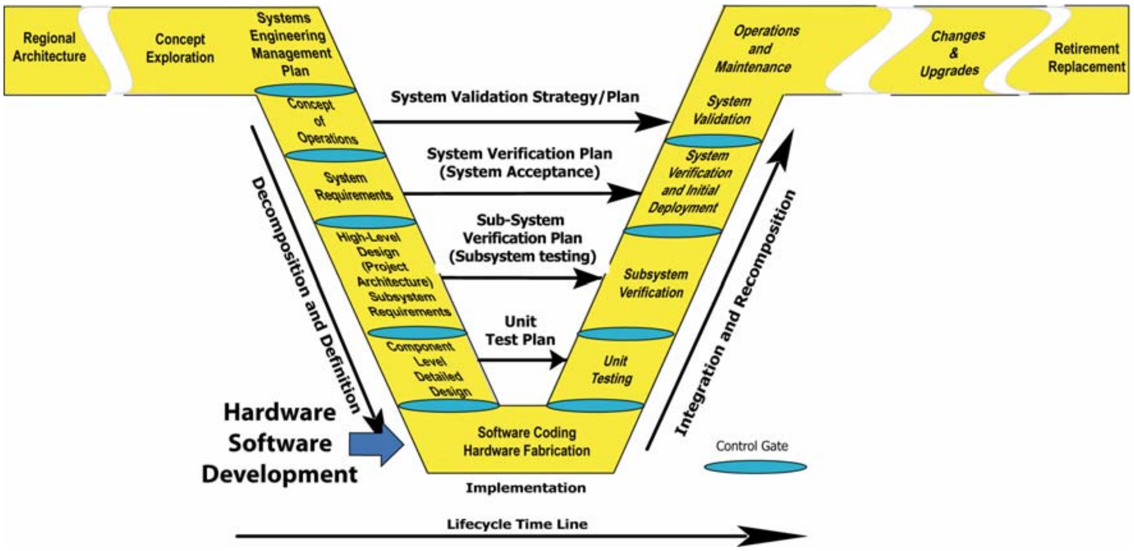
Coordinate concurrent development activities

If multiple developments are being performed concurrently based on the selected development strategy, these meetings should be coordination meetings between the developments to reduce the risk due to any integration between them. This should include schedule, functional and interface risks.

Procure COTS products

Commercial-off-the-shelf products (COTS) should be procured at this time and only if needed in this phase. If the implementation phase is planned to last several months or years, procure only those items that are needed immediately and push the procurement of this technology to the last possible minute. When doing so, account for lead times of the procurements.

The specific domain discipline e.g., software, hardware, database engineering, is expected to perform unit test, document the development environment, develop needed test simulators and perform their own developmental configuration management to the level needed to transfer the complete design package to the agency (if contracted for).



Where does the Hardware/Software Development take place in the project timeline?

Is there a policy or standard that talks about Hardware/Software Development?

FHWA Final Rule does not specifically mention general hardware/software practices to be followed. ISO/IEEE 12207 Software lifecycle process, IEEE

Which activities are critical for the system owner to do?

- Participate in the technical reviews
- Participate in risk identification and assessment.
- Participate in any project coordination meetings.
- Manage the contracting process for commercial-off-the-shelf products and applications.

How do I fit these activities to my project? (Tailoring)

Depending on the budget, staff resources, size and complexity of the project or program, the number and formality of the reviews should be tailored to fit the project. Small projects, e.g. signal system upgrades, may require only 1-2 technical reviews total and the coordination meetings will be limited to interagency or department stakeholder meetings to coordinate with communications and IT services only. Large complex projects may require bi-weekly or monthly technical reviews (at a minimum) and an equal amount of coordination meetings. The technical reviews should go in accordance with the planned reviews in the Systems Engineering Management Plan.

What should I track in this process step to reduce project risks and get what is expected? (Metrics)

On the technical side:

- During the technical reviews, a clear link must be made between the developing product and the requirement that it is intended to meet.
- During the technical reviews, the development team must show how the developing product will meet the required performance for the functionality.
- Documentation of the developed products is complete and synchronized with the detailed design documentation. Examples are code comments and artwork notes.

On the project management side:

The progress in development of hardware and software should match the planned development progress.

- For hardware developments this can be milestone-based e.g. for printed circuit boards they would be the completion of layout, artwork, fabrication, populating the board (with integrated circuits and components) and checkout.
- For software, this will be more abstract. Completion of software modules that were based on estimated lines of code developed compiled and checked out is a way to measure software progress. Another is using the function point analysis method (FPA), where completion of small program functions are measured; for example, database accesses, input/output calls, and memory access. (See Closer Look)
- Risks monitoring and corrective actions should be performed. At least once a week

project risks should be assessed per the risk management plan.

Are all the bases covered? (Checklist)

☑ Is the technical review and coordination meeting schedule established and documented?

☑ Are technical reviews conducted according to the plan?

☑ Has the development team established a schedule and method for measuring software and hardware progress?

☑ Have the significant risks been identified and is a schedule in place to monitor these risks?

☑ Does the development team have documented process for developing hardware, software, database, and communications?

Are there any other recommendations that can help?

Use an independent reviewer to assist the system owner. This independent reviewer should be technically versed and work on behalf of the system owner. This step involves a lot of technical knowledge in the specific development discipline of software, hardware, communications, and databases. An independent reviewer can help the owner of the system identify risks, completeness of design, and development performance.

It is recommended before starting implementation that the previous steps of the systems engineering process be completed. Make sure that the previous steps have been reviewed and approved by the system owner and stakeholders. This includes, in particular, that the documentation is complete.

What are the ways to estimate software development efforts?

Keep refining the software development estimates at each step of the process. And be aware of the error in software estimates. There is a lot of work being done in the software community to estimate how much effort it takes to develop major software programs. Estimating the size of a software program is done by the development team.

Each development team will have its own method of estimating code. The following are examples of methods for estimating the size of software.

- Number of requirements
- Function points
- Number of classes and objects
- Source lines of code

The following graph, adapted from Barry W. Boehm's classic Software Engineering Economics textbook 1981, shows the estimation accuracy of software efforts at different steps in the project lifecycle:

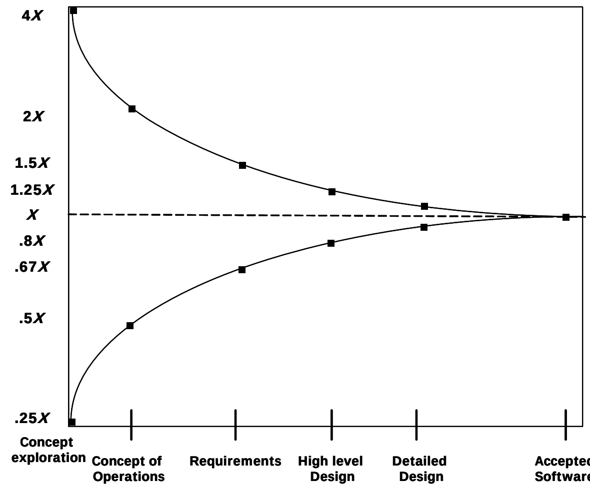


Figure 4-11 Software Estimates over the project lifecycle

As illustrated, according to the text, trying to estimate the software effort at the concept exploration phase 0 has an estimating error of plus or minus factor of 4, while estimating the software effort at the component level detailed design end of phase 2 will be much more accurate on the order of plus or minus 25%. The point is to wait until the system definition is complete before trying to estimate the software effort.

The systems engineering mindset is to <u>*push the estimation of software to the component level detailed design step of the project timeline*</u>.

In estimating software development efforts, two primary methods exist today, source lines of code and Function Point Analysis (FPA). The source lines of code has been the oldest method and a tool that is often used in this method is the COCOMO model developed by Barry W. Boehm in the late 1970's. Another method is Function Point Analysis, which also dates back to the late 1970's but has gained much popularity in recent years. Simply put, FPA estimates the number of each of five common types of program transactions that the software program will carry out and then, using other factors, such as history of function point production, estimates the software effort. Once the estimates are made, the tasks are laid out per the development plan and then monitored as part of the review process.

4.5.2 Integration

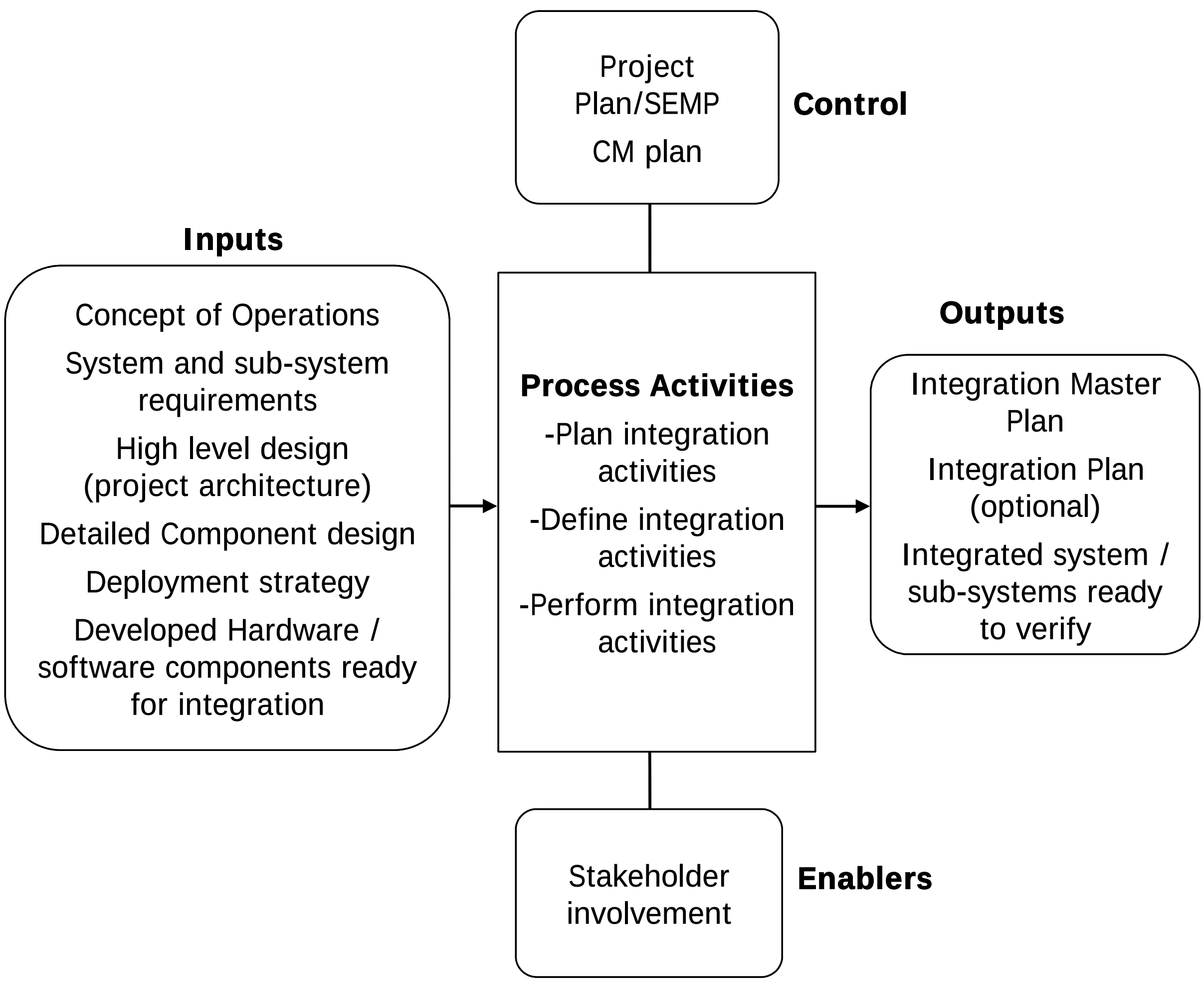
OBJECTIVE:

Integration is the process of successfully combine hardware and software components, sub-systems and systems together into a complete and functioning whole.

DESCRIPTION:

Integration is an iterative process, taking hardware and software components and forming them into complete sub-system elements and then taking the sub-system elements and combining them into larger combined sub-systems until all sub-systems are combined into the final system. Integration planning starts when the project activities are first defined. The next major input occurs when the sub-systems are identified during the high level design and project architecture step. Finally integration is performed when the hardware and software components are developed and delivered by the development team. Integration and verification are closely linked processes in which one follows the other until the entire system is ready for operational deployment. A complex project may need a written Integration Plan. Integration activities are driven by system requirements, internal interfaces within the system and external interfaces to legacy systems and the deployment strategy. Integration activities are performed iteratively along with verification.

CONTEXT OF PROCESS:



INTEGRATION PROCESS

Inputs:

Concept of Operations describes the way the system is to operate and will assist in the verification and integration effort.

System and Sub-System Requirements contain the requirements for the sub-systems / systems

High Level Design (project architecture) defines the integration activities to be performed.

Component Level Detailed Design contain the design constraints for the sub-systems / systems to be integrated

Deployment Strategy defines when and where the sub-systems are to be grouped and deployed

Developed hardware / software components and sub-systems have completed integration and are ready for the next level of verification.

Control:

Project Plan/SEMP establishes a high level description of the systems engineering plan for integration.

Configuration Management Plan sets the configuration controls needed during integration.

Enablers:

Stakeholder involvement is needed to assist with integration with external systems and devices

Outputs:

Integration Master Plan establishes the goals and high level approach to integration

Integration Plan (optional) documents the high level plan and process for integrating the system this is part of the Project Plan/System Engineering Management Plan (SEMP)

Integrated sub-systems / system ready for verification

Process Activities:

Plan integration activities:

Planning includes the sequence in which the various components of the system should be integrated. The needed resources, schedule, and coordination activities, if multiple development teams are involved, and finally the documented plan itself. A number of factors influence the integration sequence, including:

The order in which components and sub-systems are produced by the development team(s).

Each integration step should produce a product that implements a related set of functionality. For example, an operator interface may be integrated with a loop data collection function before the loop data function is integrated with an incident management function.

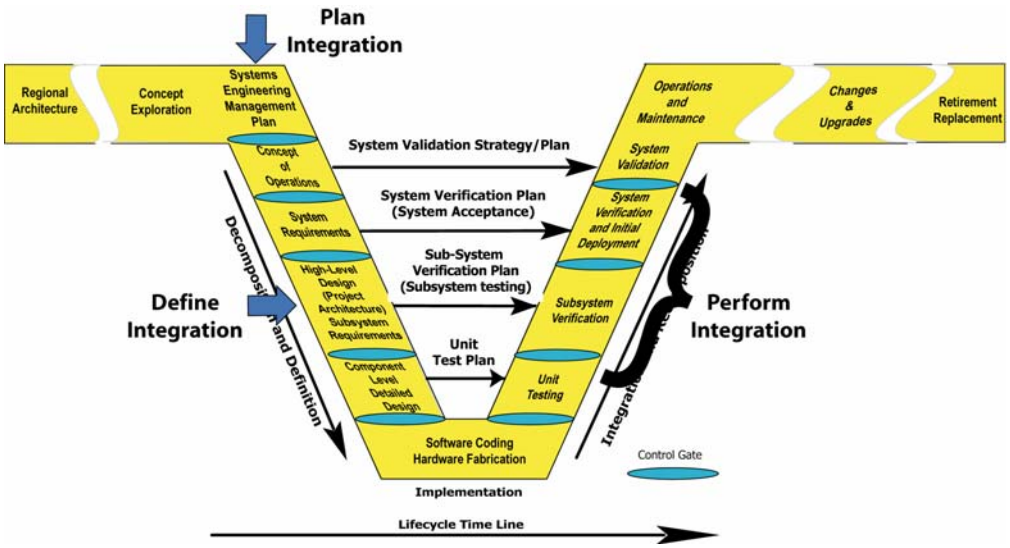
Define integration activities:

At the high level design (project level architecture) integration activities are defined. Sub-systems, internal interfaces, and external interfaces are defined and are the key points for integration. Also at the high level design, the number of integration / verification cycles is defined.

Perform integration activities:

The first step is to ensure that the integration team has access to the resources needed to support the planned integration step. Special attention has to be paid to resources that come from outside the organization the integration team is in. These could include: support from the developers or manufacturers, support from other agencies with an external interface, a testing environment (e.g. workstations, communications, and interface simulators.) and, of course, the various sub-systems to be integrated.

As integration proceeds, progress is monitored through periodic reviews. These reviews check progress according to the schedule, ensure that problems found during the process are being resolved, that requirements changes, if necessary, are being made according to the Configuration Management Plan, that needed resources are available or that a suitable substitution is found, and that the integration effort is thorough enough to support subsequent verification. As the cycle of integration and verification is repeated, lessons learned during a verification step may have to be fed into the next round of integration. Integration should be complete enough that subsequent verification proceeds with minimum disruption.



Where does Integration take place in the project timeline?

Is there a policy or standard that talks about integration?

FHWA Final Rule does not specifically mention integration as one of the required systems engineering analysis activities. EIA 731 and CMMI have identified best practices for integration.

Which activities are critical for the system owner to do?

- Determine the need for a written Integration Plan
- Review and approve the Integration Plan, if one is needed
- Manage the timely acquisition of resources needed to support integration
- Track the progress of integration with respect to the project schedule and intervene if the progress falls behind the schedule

How do I fit this step to my project? (Tailoring)

There are a number of factors which will make a project complex. The same factors that influence other steps in the systems engineering process also influence the integration process. Integration of sub-systems and integration with external interfaces is always required and can't be avoided. The major variable in the integration process is the degree of formality and organization it needs. The simpler the system, the smaller the project team and the fewer the number of external stakeholders (stakeholders with systems that interface with your system), the less formal the integration process needs to be. If the people involved can coordinate their effort without the need for a written Integration Plan, but just following the more general guidance of the Program Plan and the SEMP, then substantial effort can be avoided by not preparing such a plan.

What should I track to reduce project risk and to get what is expected? (Metrics)

The number of times failures are detected during integration is a good indicator of the quality of the development effort.

The number of times a later stage of integration turns up a problem that should have been detected in an earlier stage of integration is a good indicator of the quality of the integration effort.

The number of times problems are not found in integration but are discovered during verification is another good indicator of the quality of the integration effort.

Are all the bases covered? (Checklist)

☑ Are integration activities included in the master project schedule?

☑ Does the plan for integration and verification support the strategy for deployment?

☑ Based on project complexity, is a written Integration Plan required?

☑ Are the external systems needed to support integration available, or does the interface need to be simulated?

☑ Have the components to be integrated been placed under configuration control?

☑ Are the development teams available to promptly fix problems uncovered during integration?

Are there any other recommendations that can help?

The importance of a good strategy and verification of design:

Develop a good integration strategy: A successful integration process is based on a sound strategy that will give it direction and completeness. This same strategy will be needed to guide the verification and initial deployment activities. This strategy is based on a set of goals that were established early in the planning stages of the project. These goals answered the following questions:

- In what order do you need to deploy these capabilities in order to provide useful operational capabilities at each step?
- How do you want to evolve the operational capabilities at a location in order to provide increasingly useful operational capabilities?
- What are your funding limitations?

Of course that last goal, spending the available funds in the most effective manner, is usually the hardest to solve. Since these goals are related to deployment, this subject will be revisited in that section. Nevertheless, your integration plan, as well as your design, has to be fashioned to meet these deployment goals.

As has been stated before, integration is a much more informal activity than verification. As such, the preparation of detailed plans and procedures is generally not required. In fact, if such structure is felt to be necessary, the procedures used for subsequent verification can also be used as part of the integration activity. Thus the verification dry-run (see the verification section) could also be seen as part of the integration effort.

Verification of design:

Integration is more than a verification of requirements; it is also a verification of the design. It explores the details of both the hardware and software. It needs, for instance, to look at hardware and software interfaces at a much lower level than just exercising the functional requirement.

Generally this informal integration approach is effective as it avoids the costs of more formal documentation. Still, it needs to be carefully monitored, it needs adequate project support, and it needs the right people on the integration team.

A closer look at integration and verification* and *levels of integration

Integration and verification are iterative processes with each other. Integration puts a sub-system (or system) together (from components and/or other sub-systems) and informally assures that everything is working as it should according to the requirements. Verification much more formally tests the assembled system (or sub-systems) to show that all applicable requirements are met. The figure shows this cycle and how it is repeated until the entire system can be accepted.

Levels of integration means that the levels of integration needed for a system will match the number of levels of the system hierarchy. For example, a traffic control system having 3 levels of hierarchy would include the following: component level (the loops and field controllers) would be the first level of integration; the sub-system level (field controllers with field masters) would be the next level of integration; and finally the system level (host with the field masters and field controllers) would be the final level integration. More complex systems may have additional levels of hierarchy and integration. For example, in regional Intelligent Transportation Systems, the traffic control system example above may need to integrate with a freeway ramp metering system for coordination. This would represent a fourth level of integration and so on.

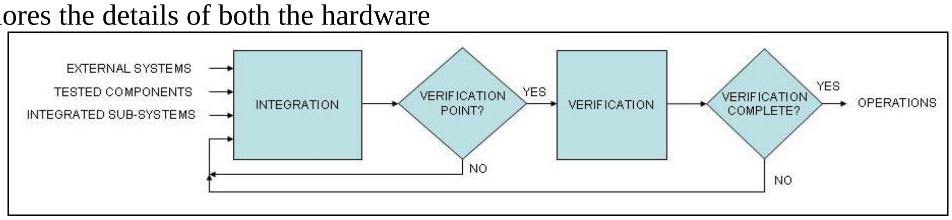


Figure 4-12 Integration and Verification are Iterative

4.5.3 Verification

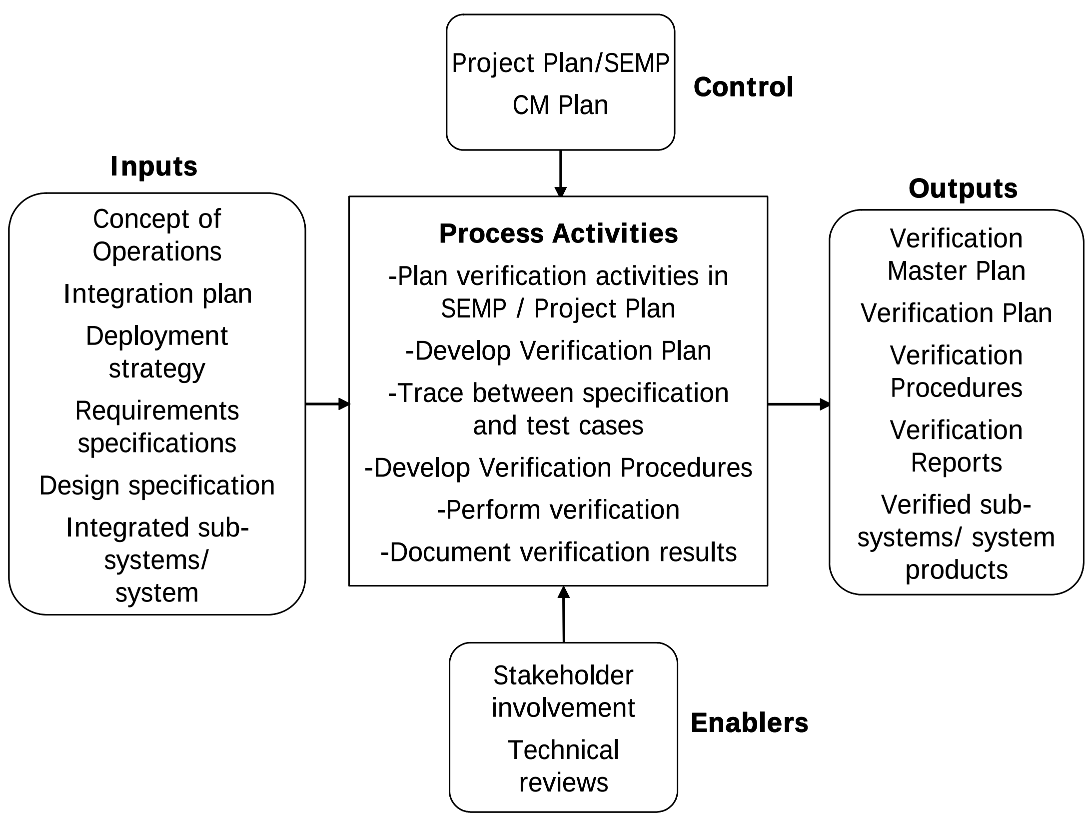
OBJECTIVE:

The verification process is used by the system owner and by other stakeholders to show that the as-built system, sub-system and components meet all of their requirements and design. This process is used by the system owner and other stakeholders to accept the system products from the development team.

DESCRIPTION:

Verification is the process that proves the system (or sub-system or component) meets its requirements and matches the design. Since verification is based on requirements and design, one of the keys to successful and effective verification is well-written and complete requirements and design documents. These requirements and design elements are developed, reviewed and approved earlier in the project timeline before the system is developed or procured. Planning for the verification activities starts with the System Engineering Management Plan (or with the Project Plan if a SEMP is not needed). At this level the general structure of the verification tasks is identified and shown to be compatible with the desired deployment plan and with the system concept. The Verification Plans are best written at the same time the requirements of the system, sub-system, or component are developed, mainly to show that the requirements, as written, can be verified. At the end of the detailed design effort, verification procedures can be written. These procedures are the detailed steps to be taken to verify each requirement and design element. There must be a clear trace from each requirement, through the Verification Plan, down to a detailed step in the verification procedure. Verification is performed iteratively with the integration activities starting at the component level and progressing through sub-system development to verification of the entire system. Final verification for system acceptance is done with the installed system. At this point system development is complete and the deployed system is ready for operations. Stakeholder involvement, including by the system owner, is critical in verification.

CONTEXT OF PROCESS:



VERIFICATION PROCESS

Inputs:

Concept of Operations describes the way the system is to operate and will assist in the verification and integration effort.

System and Sub-System Requirements contain all the functional and performance requirements to be verified

Design Specifications contain the design elements to be verified

Integration Plan (optional) shows how the integration steps are to be done iteratively with verification

Deployment Strategy (optional) defines when and where verification takes place

Integrated sub-systems/system are ready for verification

Control:

Project Plan/Systems Engineering Management Plan (SEMP) establishes a high level description of the project's plan for verification

Configuration Management Plan sets the configuration controls needed during verification

Enablers:

Stakeholder involvement is needed for verification conduct and to show critical stakeholders that the system meets its requirements

Technical Reviews include a test readiness review to determine that all resources needed for a verification step are available

Outputs:

Verification Master Plan is included in the Project Plan/SEMP to establish the general guidelines for this important part of the systems engineering process

Verification Plan documents the plan for verifying the system and sub-system requirements

Verification Procedures document the details of each verification step

Verification Reports document the results of each verification step

Verified sub-system/system ready for further integration, deployment or operational use

Process Activities:

Plan verification activities in SEMP / Project Plan

During the project planning stage a strategy for verification is developed that is compatible with the system concept and the deployment objectives.

Develop Verification Plan

Verification Plan is written for each level (component, sub-system or system). The plan will establish a test case and verification technique for each requirement and for each design element contained in the applicable Requirement Specification.

In addition to the test cases, the Verification Plan will give general guidance for all of the verification activities. These include the Identification of all verification participants and description of their roles and responsibilities, a schedule for verification activities, and finally the identification of test equipment needed and of software drivers or simulators needed to model the interfaces to the system under test.

Trace between specifications and test cases

Each test case is traced to a specific requirement to ensure that all requirements are verified.

Develop Verification Procedures

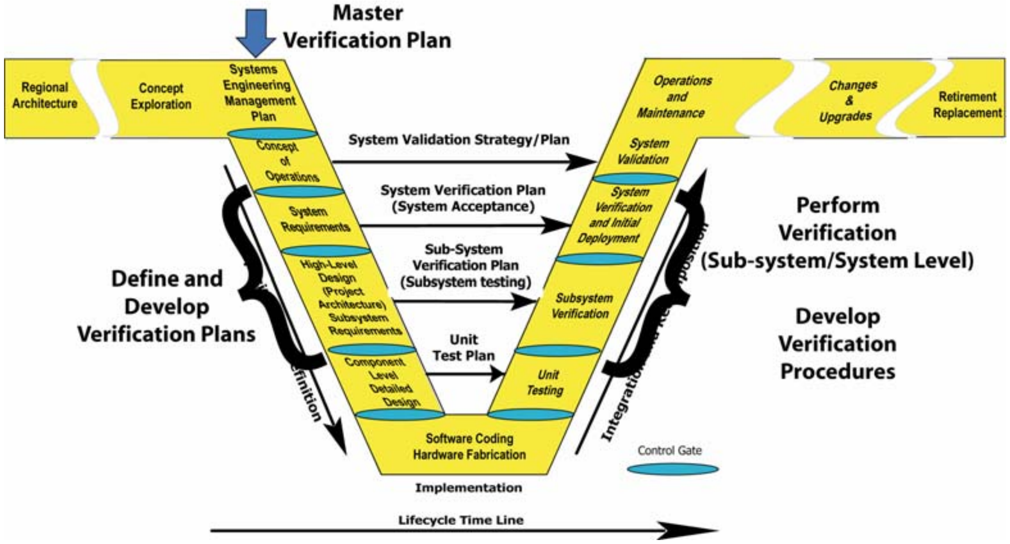
These procedures are the detailed step-by-step actions and expected outcome for each verification test case.

Perform verification

When all needed resources are ready, verification is performed according to the approved procedures.

Document verification results

Prepare a Verification Report for each verification step.



Where does Verification take place in the project timeline?

Is there a policy or standard that talks about verification?

FHWA Final Rule does not specifically mention general verification of requirements it does require interoperability tests relating to use of ITS standards. IEEE std. 1012 talks about independent verification and validation. CMMI identifies best practices.

Which activities are critical for the system owner to do?

- Identify and recruit stakeholders who are needed to participate in verification
- Review and approve all documents
- Witness critical verification steps

How do I fit this step to my project? (Tailoring)

Some level of verification is needed to accept the system. It is the formality with which verification is performed that can be tailored to the budget, size and complexity of the project. For a small simple project with few stakeholders, it only may be necessary to use the requirement document itself as a checklist and extemporize the procedures on the fly. Thus no verification documents are needed. The system owner determines what level for verification formality and documentation is needed to satisfy the complexity of and stakeholder participation in their project.

What should I track to reduce project risk and to get what is expected? (Metrics)

Number of verification failures and their cause (poor requirements, design errors, inadequate integration) is an indication of the quality of products from the development team.

Are all the bases covered? (Checklist)

- ☑ Did you prepare and review a Verification Plan?
- ☑ Did you trace all requirements to a Verification Plan test case?
- ☑ Did you prepare and review a Verification Procedure?
- ☑ Did you identify and train all participants?
- ☑ Did you determine the readiness of all resources needed for testing?
- ☑ Did you notify participants of the testing schedule?
- ☑ Did you prepare a Verification Report?

Are there any other recommendations that can help?

A closer look at the stages of verification, verification techniques and the rules for performing verification

Key stages of verification

A project may require three or more different stages of verification: sub-system, system, and commissioning. The first is iterative with integration and the last is iterative with deployment. System verification and acceptance falls between the two. Of course, special project situations may require some tailoring and perhaps additional stages for complete verification.

- Sub-system Verification – As discussed in 4.4.2, High Level Design, a system is often divided into two or more sub-systems for ease of development. Once the integration process has produced one of these sub-systems, it is verified against its requirements. Once verified, the sub-system can be integrated with other sub-systems.

- System Verification – This step covers all integrated sub-systems and is usually used to accept the entire system. For many requirements this is the last time they are verified. As such, this step is the most formal, the most reviewed, the most witnessed and where failures receive the most attention. Although it may not be as exhaustive as sub-system verification, it still must be extensive enough to produce a solid feeling among your stakeholders that the system does what it is supposed to do.
- Sub-system and system verification is best done in a highly controlled environment, especially with respect to external inputs to the system under test. This usually requires software to simulate, or model, the external world. For instance, a traffic signal simulator or roadway sensor simulator may be needed to test a new central control system.
- Commissioning – Commissioning is done after the system is deployed, to verify the system works when installed. Commissioning is generally much more cursory than system verification, just enough to verify that everything is still working. However, in some circumstances, a part of system verification must be deferred to the time of commissioning. For instance, maybe only simulated real-world inputs from critical sensors are available for system verification.
- Verification of the system's ability to work with the complete set of real sensors must wait until after deployment.

You may find it necessary to overlap the last two stages of verification. System verification can be started in a development environment using simulated inputs from sensors and external system then completed after deployment and commissioning using real sensors and real external systems. While verification with simulated inputs may be necessary, final verification with real inputs is almost always mandatory.

Verification techniques

Four techniques are used to verify requirements: inspection, analysis, test and demonstration.

- Inspection – Visual verification of a requirement, such as a color, a size, and model number.
- Analysis – Mathematical analysis of collected data to verify a requirement
- Demonstration – Use of the system itself to verify the expected output, such as a response to an operator input. This is the most commonly used verification technique.
- Test – Similar to a demonstration except external test equipment is used.

A special type of demonstration is called a burn-in is used to identify and resolve random or latent defects (thermal or memory leaks).

General rules for performing verification

- Notification of all stakeholders of the schedule for verification and clarification of their roles and responsibilities
- Verification of the configuration of the system under test
- Process for recording all test actions and the system's response
- Process for dealing with all unexpected responses, including:

  Verifying and recording the unexpected response

  Team analysis of the reason for the unexpected response (e.g. procedure error, procedure not followed, or system failure.)

  Team agreement on a plan of action based on this analysis. (e.g. repeats the test, revise procedure, change the requirement, suspend the test, fix the system and retest)

Be careful of requirements creep. During verification some stakeholders, especially if they have not been involved in the design activities, will want to rewrite or add to the system requirements. A typical example is a desire to change the operator interface. You can't let this happen, because at this stage it is most expensive. The best way to avoid this is to ensure that the right and complete set of stakeholders is involved in establishing the requirements and designing the system from the start.

4.5.4 Initial System Deployment

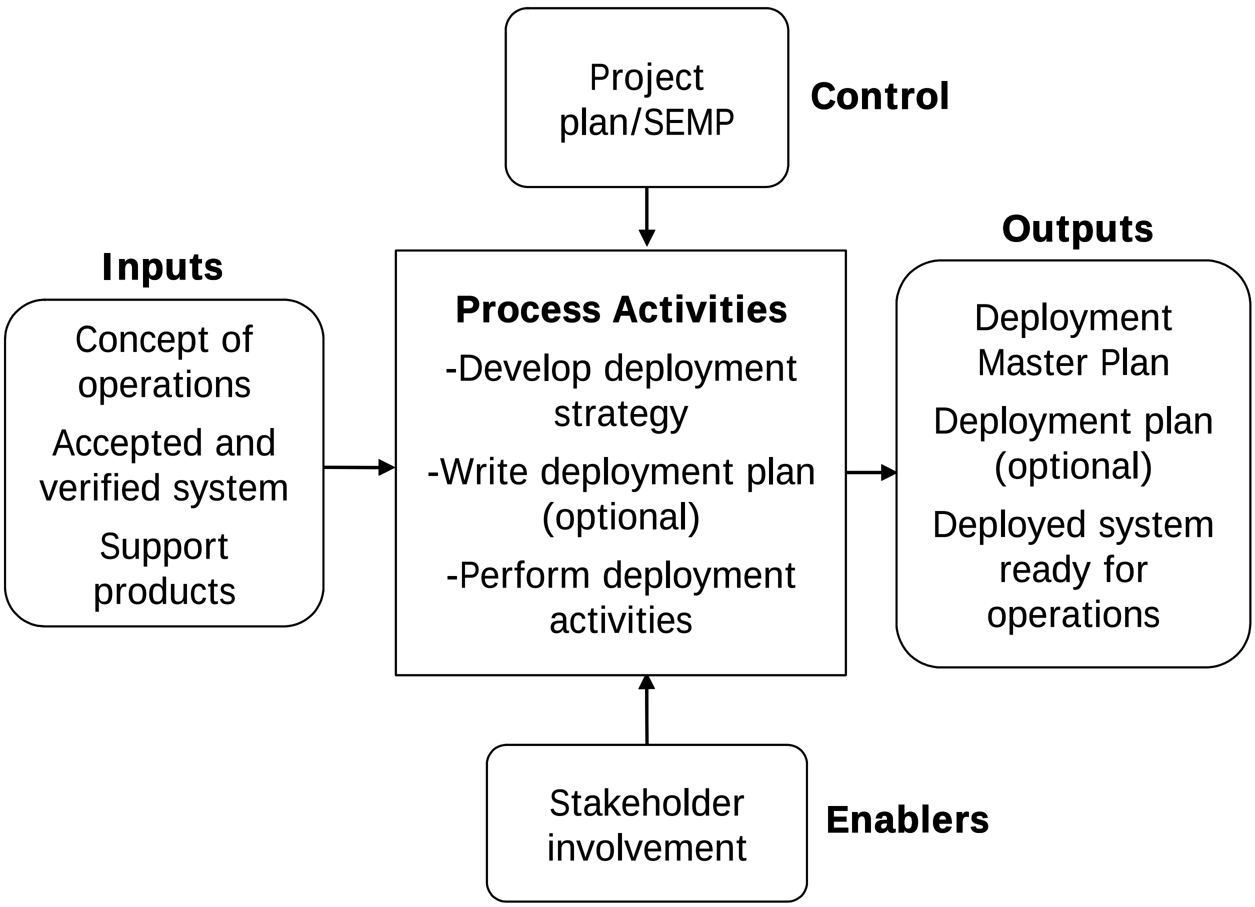
OBJECTIVE:

The deployment process takes the tested subsystems and installs them in preparation for operations.

DESCRIPTION:

Deployment is the final design/build step in the development of a system. The deployment strategy must reflect the plan for the project and provide an operationally useful component of the system at each step of the process and at each deployment location. The deployment strategy may involve a single deployment to a single site or may have to deal with multiple partial deployments to multiple sites over an extended period of time. A complex deployment also may require post acceptance testing at each site. A written Deployment Plan may be necessary to ensure a successful deployment, especially if multiple agencies are involved. A Deployment Plan will define all the work steps for complete deployment and who does them. At each deployment site the hardware and software is configured, installed and then tested to show it is ready to support operations.

CONTEXT OF PROCESS:



INITIAL SYSTEM DEPLOYMENT PROCESS

Inputs:

Concept of Operations provides general guidance on how the system is to be operated and therefore on how it must be deployed

Accepted and verified sub-systems / system are ready for deployment

Support Products includes training materials, users, and maintenance manuals.

Control:

Project Plan/SEMP establishes a high level description of the project management and system engineering plan for deployment.

Enablers:

Stakeholder involvement is needed to support the deployment activities

Outputs:

Deployment Master Plan establishes the goals and a strategy for deployment. This is included in the Project Plan/System Engineering Management Plan (SEMP)

Deployment Plan (optional) documents the high level plan for deploying the system

Deployed system is ready for operational use

Process Activities:

Develop Deployment Strategy

The strategy defines what capabilities and parts of the overall system will be deployed where and when. The Strategy is used to allocate funding for the project over time by identifying what the timeline will be for the projects.

Write Deployment Plan (optional)

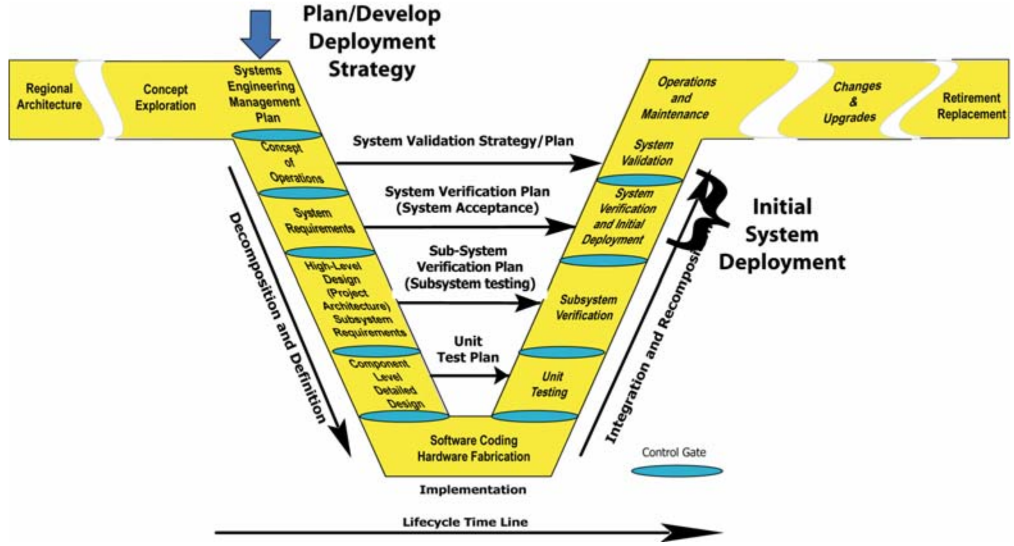
A number of considerations may lead you to prepare, review and approve a written Deployment Plan. These include:

- A complex deployment schedule with multiple deployments, of different configurations, to multiple sites. For instance, you may be deploying a number of Transportation Management Systems statewide with different configurations at each site.
- The needed facilities such as electrical, air conditioning, communications infrastructure, and lighting needed to support the system. In addition, personnel training will be needed for operations and maintenance. This must be planned and performed in time for the delivery of the system.
- Several stakeholders whose activities must be coordinated for the deployment effort, especially stakeholders from different organizations and agencies. For instance, even a single ITS site may have multiple interagency interfaces that, when implemented, will change the operations at these external systems.
- The need to sell your selected deployment plan to management by showing the analysis of alternatives that led to the selective approach. This is especially useful if you are trying to balance operationally viable deployment steps with a restrictive funding availability.

Whether or not a written Deployment Plan is needed, the planning must consider what parts of the system, with what capabilities, are deployed where and when.

Perform deployment activities

Managing deployment follows the same path that integration and verification have followed. First, all needed resources must be identified, obtained and trained, including all facilities (electrical, communications, lighting) and personnel training for operations and maintenance. Then, just prior to the start of each deployment step, the readiness of those resources is determined and any work-around plans put into effect. During the performance of a deployment step, progress should be monitored and reviewed with the deployment team on a regular basis. The final step of a deployment is usually an integration and verification of the deployed system prior to operational acceptance.



Where does Initial System Deployment take place in the project timeline?

Is there a policy or standard that talks about deployment?

FHWA Final Rule does not specifically mention initial system deployment as one of the required systems engineering analysis activities.

Which activities are critical for the system owner to do?

- In concert with the operating agencies, develop, review and approve the goals and a general strategy for deployment
- Identify and recruit agency stakeholders to participate in deployment
- Review and approve all deployment plans
- Monitor deployment activities and witness critical post deployment verification

How do I fit this step to my project? (Tailoring)

Depending on various factors of the project, deployment can range from very simple to very complex. The number of deployment steps and the number of stakeholders involved in deployment are the best indicators of complexity, although there may be others of equal importance. If either of these factors warrant, then project management may decide that the expense of preparing, reviewing and approving a Deployment Plan document is justified. If it is not, then the guidance in the Program Plan and in the SEMP, plus a qualified person in charge of deployment, is quite sufficient.

What should I track to reduce project risk and to get what is expected? (Metrics)

Deployment involves elements of both integration and verification and each of these processes has its own set of useful metrics. Beyond that, tracking progress to the schedule is the most useful thing project management can do to reduce project risk and get what is expected.

Are all the bases covered? (Checklist)

☑ Have you developed a comprehensive set of deployment goals that meet the stakeholder needs?

☑ Can those deployment goals be traced into the deployment strategy?

☑ Does the deployment strategy consider available funding?

☑ Does each step in the deployment strategy produce an operationally useful and maintainable deployed system?

☑ Does the deployment strategy minimize the risk of interference to on-going operations?

☑ Does the deployment strategy offer a viable operational fallback at each step of the process?

☑ Are all stakeholders in a deployment step aware of their roles and responsibilities?

☑ Are all resources needed for a deployment step available?

☑ Do you have a work-around plan in case a needed resource is not available?

Are there any other recommendations that can help?

Factors that should be considered when developing a deployment plan:

- If multiple locations are involved, the final desired configuration at each site
- If multiple sites are involved, the relative sequence in which each site needs to reach its desired final configuration
- The dependence on prior deployments to this or any other site. For instance, an operational site only may be viable if a maintenance center needed to support the operational site has been previously upgraded or installed.
- If a phased deployment is required (say due to a funding profile spread over several fiscal years) then a number of other factors must be considered, including:
    - Each incremental deployment phase must result in an operationally useful system
    - Each incremental deployment phase and all dependencies must be included or already installed (it does little good to install capability B if capability A is needed to use B but A is not installed until later.
    - The cost of each incremental deployment phase cannot exceed the incrementally available funds

<u>Using the Deployment Plan</u> for selling the strategy and to provide planning and advice for a <u>"ribbon cutting" ceremony</u>

Use the Deployment Plan document to "sell" the selected deployment strategy. This is much more likely when a relatively complex set of deployment goals have to be met, such as when the conflicting goals of operationally useful but funding-constrained deployment phases are required. It then becomes necessary to show that not only the selected strategy meets those goals but meets them better than any alternative strategy. The Deployment Plan is then an excellent place to document this strategy selection since much of the information is eventually needed for implementing the deployment plan.

<u>Plan for the "ribbon cutting" ceremony</u>. Since the deployment activity is the last step in the development process and the point where the system is turned over to the system owner, there is sometimes a desire to turn that hand-over into a "ribbon cutting" ceremony. If this, or any other public relations type of activities are required of the project office (as opposed to being the responsibility of the operating organization), then planning for this activity should be included as part of the deployment effort and, if one is written, documented in the Deployment Plan.

Make sure that the operational and support team is in place when the system is commissioned into operations. In addition to the challenge of deploying an operationally viable system that meets all of its requirements, very often two other conditions have to be met. The first is that the operations people have to be available and trained in the new system's features. This may involve the recruitment of additional staff and certainly includes operational training for both new and existing staff. The second condition is to ensure that adequate maintenance support will be available. Not only does this require trained staff, but also sometimes additional facilities are required. Sometime an existing maintenance facility has to be upgraded with additional test equipment and additional spare parts to support your system's new hardware. Sometimes a software test bed has to be created to give support staff a place to fix and test the existing software products and to develop upgrades to those same products, without interfering with normal use of the operational system.

4.6 Validation, Operations and Maintenance, Changes and Upgrades

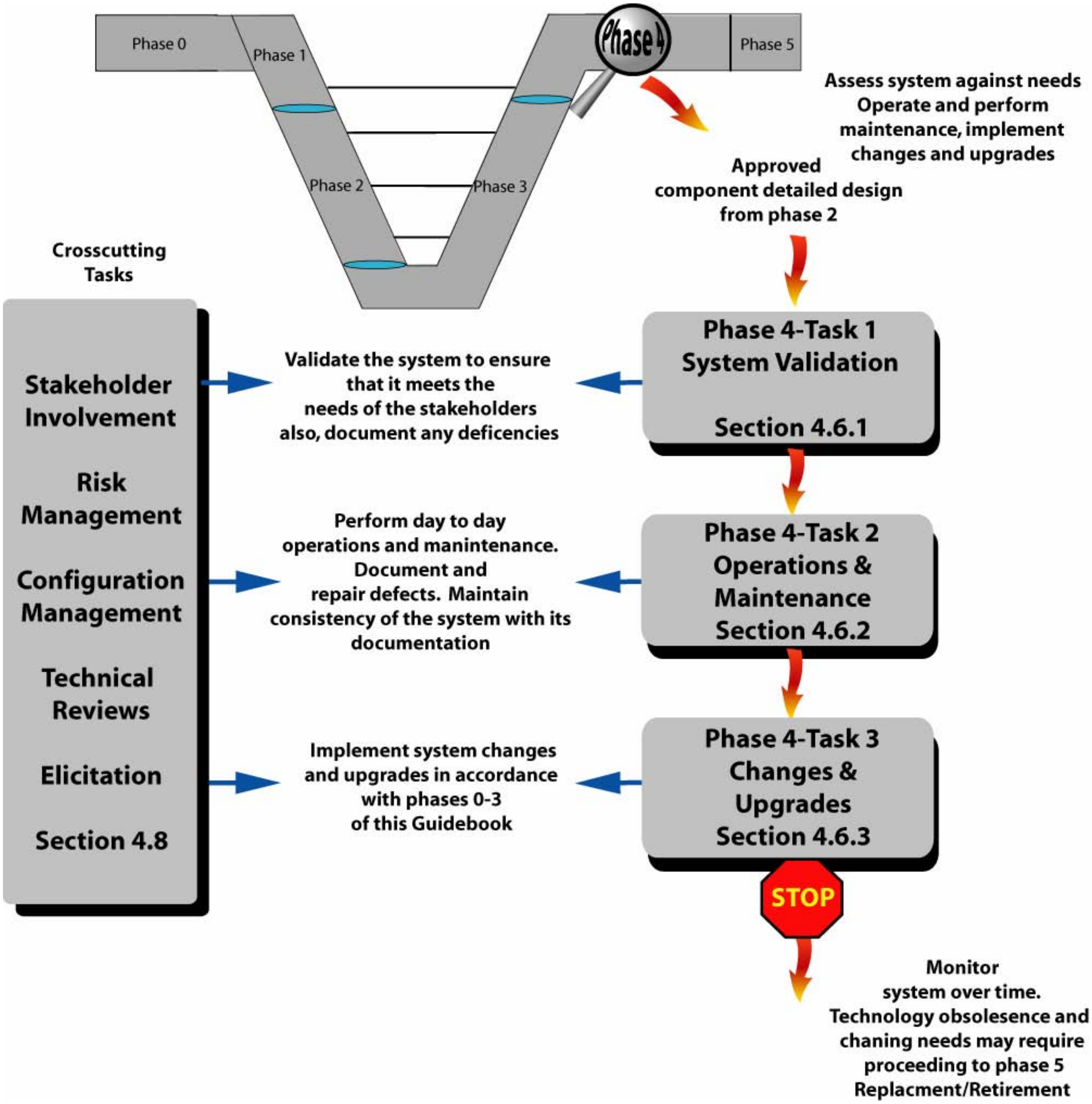


Figure 4-13 Phase 4 - Validation, O&M, Changes and Upgrades Roadmap

4.6.1 System Validation

OBJECTIVE

Validation is an assessment of the operational system. Validation is used to see that the system meets the intended purpose and needs of the system owner and stakeholders.

DESCRIPTION:

Validation starts with a clearly stated set of needs. These needs are the basis for the system requirements. When the system is developed the system is assessed against these needs.

The validation process has three primary activities:

Planning – With stakeholder involvement planning starts at the beginning of the project timeline. The plan includes who will be involved, what will be validated, what is the schedule for validation, and where the validation will take place.

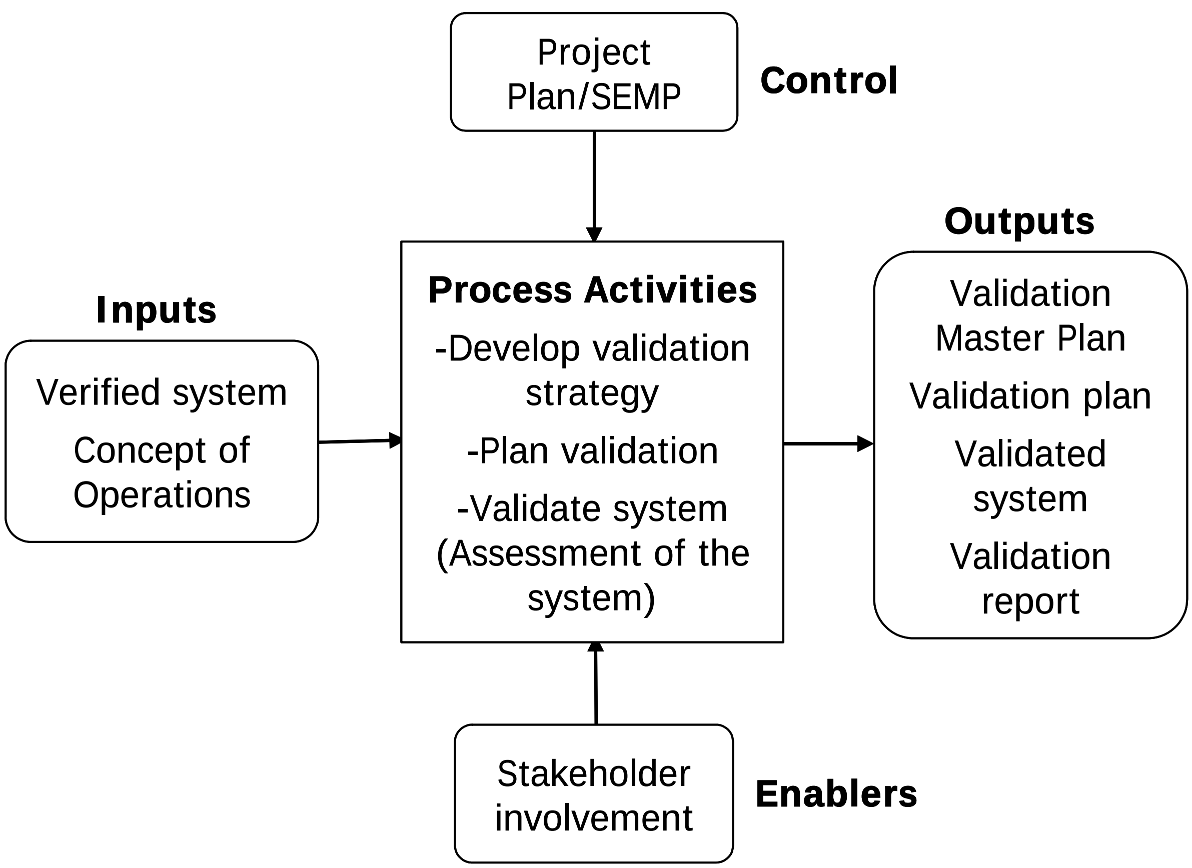
Validation strategy – This defines how the validation will take place and what resources will be needed, for example whether a before and after study will be needed; if so, the before study will need to be done prior to deployment of the system.

Perform validation – After the system has been accepted, the system should be assessed based on the planning and strategy and the results documented.

The system owner and stakeholders are responsible for the validation of the system. The primary systems engineering activity is to assist in the development and execution of all three activities.

Validation answers the question "Was the 'right' system built?"

CONTEXT OF PROCESS:



SYSTEM VALIDATION PROCESS

Inputs

Verified system – After the system has been verified (accepted by the system owner) it is then ready for validation testing.

Concept of Operations provides the goals, objectives and needs to be assessed.

Control

Project Plan/Systems Engineering Management Plan (SEMP) includes the validation plan that will be used identify the strategy, schedule, and resources for validation.

Enablers

Stakeholder involvement includes not only the system owner but its stakeholders as well. Each will have needs that the system is intended to address. When the assessment is performed the stakeholders must be in agreement on the plan, strategy and outcome of the assessment.

Outputs

Validation Master Plan specifies what needs to be validated, where and when. This becomes part of the Systems Engineering Management Plan (SEMP).

Validation Plan defines how the validation will be performed. In particular, it specifies whether a before and after study is needed or if special environmental conditions or resources are needed or resources needed to conduct the assessment.

Validated system (Assessment of the system) is one that has been assessed against the initially stated needs. It may have fallen short in some areas and exceeded in others. The short falls are used to identify new requirements for the evolution of the system.

Validation report documents the results of the validation process and the strengths and weaknesses of the system and where improvement can be made.

Process Activities:

Develop validation strategy

Validation planning occurs at the beginning of the project and is part of the Systems Engineering Management Plan. The plan includes the environment for validation resources. A validation plan is developed as part of the systems planning and concept of operations.

Plan validation

Strategies include alpha, beta testing and evaluation period for validation. If before and after studies are needed it will be identified in the strategy.

Validate system (Assessment of the system capabilities in operations):

Once the system has been accepted and deployed, the functionality and performance of the system are validated (assessed) against the needs, goals, and objectives as stated in the concept of operations. Also, the system is assessed in the "real-world" operations to evaluate the system against expectations of the system owner and stakeholders. This evaluation can result in one of the following:

Case 1) System performs as expected.

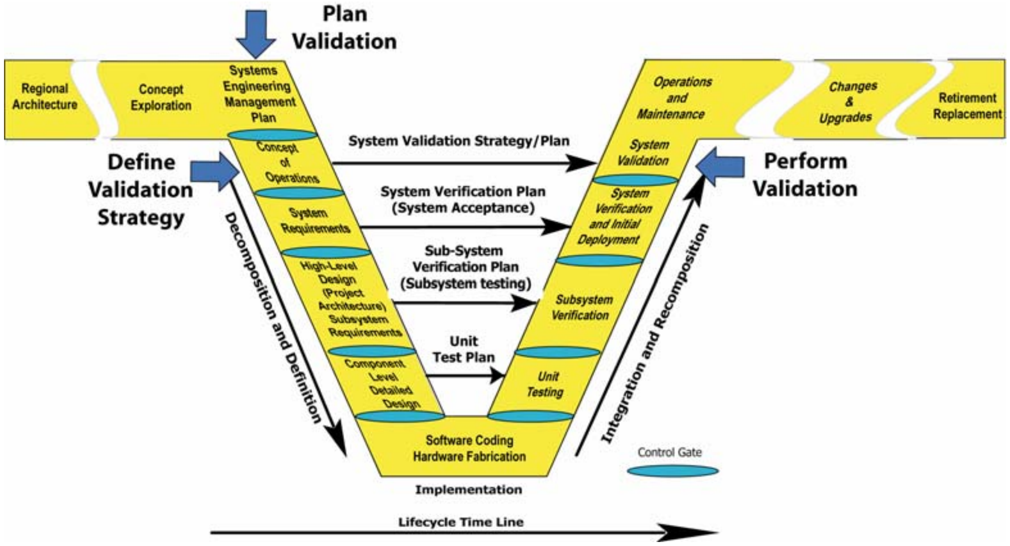
Action: Expand the system to address additional needs and document the emergent qualities of the system as it is in operations. New requirements will be developed for the next evolution of the system.

Case 2) Needs were not clearly articulated and the system falls short of the expectations.

Action: Improve the process used for the elicitation of needs and involvement of stakeholders and then correct the definition of needs. Develop the correct set of requirements for the next evolution of the system.

Case 3) The problem space was not understood and the needs were based on the ill-defined problem to be solved.

Actions: Improve the problem definition process and the elicitation processes. Re-evaluate the problem space and needs to ensure that it is understood for the next evolution.



Where does the Validation take place in the project timeline?

Is there a policy or standard that talks about Validation?

FHWA Final Rule does not specifically mention general validation practices to be followed. IEEE-1012 Independent verification and validation and CMMI identifies best practices

Which activities are critical for the system owner to do?

- Lead in developing the plan, strategy and performing the validation of the system.
- Gain stakeholder involvement in the validation process and gather their expectations for the system and performance outcomes.
- Participate in requirements walkthrough and make sure the correct requirements are being developed (Validating the requirements)

How do I fit these activities to my project? (Tailoring)

There is great latitude in system validation. It is dependent on institutional agreements, State and FHWA requirements on a per project basis. In signal upgrade systems a simple before and after study on selected intersections may be sufficient to validate. In a more complex system a number of evaluations may be needed. This validation may be needed for each stakeholder element, each subsystem e.g. camera, CMS, and detection system. It may be done on a sample area of the system or comprehensively. Getting this addressed with the stakeholders in the planning stage will be very important.

What should I track in this process step to reduce project risks and get what is expected? (Metrics)

On the technical side:

- Each need, goal and objective should have an element that can be measured and tracked throughout the development. For example, for an incident management system the goal of the planned systems may be to reduce incident management time by 30%. The technical metric is "time", which includes for example, detect time, time to verify, response time, and time to clear. The time would be the metric to monitor throughout the development.

On the project management side:

- At this point the development is complete but, as the project manager, it will be important to validate the systems as soon as possible and in accordance with the plan. If validation is delayed too long, the assessment may become much more difficult to accomplish (lack of resources and interest) and with the changing environment the results of the assessment may become diluted. (E.g. change in traffic patterns, increase in congestion over time).

Are all the bases covered? (Checklist)

☑ Were all the needs clearly documented?

☑ With each need, goal and objective is there an outcome that can be measured?

☑ Are all the stakeholders involved in the validation planning and the definition of the validation strategy?

- ☑ Are all the stakeholders involved in the performance of the validation and agreement reached on the planned outcomes?
- ☑ Are there adequate resources to complete the validation?
- ☑ Are the system owner and stakeholders participating in the requirements walkthrough and approval process?
- ☑ Is there adequate systems engineering support for the validation planning, strategy and performing validation?

Are there any other recommendations that can help?

This is an area of high stakeholder involvement. Ample time should be given to this activity. Clearly identifying measurable needs, goals and objectives is critical for assessing the system as well as the development of a good set of requirements for the system.

The systems engineering mindset is to "start at the finish line" (what the system is to do and how well it is to do it). This clear end point is essential for successful completion of the system. The journey may encounter detours, road blocks, and it may be longer than expected. The validation process helps the system owner in making this "finish line" clear to the stakeholders and to the development team.

Validate the system as quickly as possible. There may be a tendency to lose interest once the system has been developed, accepted by the system owner, deployed and commissioned into service, assuming that the system is doing the job it was intended to do. *With Intelligent Transportation Systems (ITS) it is not only the delivery of the project (system) that is important, but that the* ***"right"*** *project (system) was delivered. This can only be done through the validation process.* The system owner and stakeholders should follow through as soon as possible with the assessment of the system.

What is the difference between *Validation and Verification?* First let us look at validation, then verification.

In general, validation determines if the correct or right system is being developed. This means that the system under development *will meet the intended needs of the system owner and stakeholders* when completed. Does the system solve the problem or issue that it was intended to solve? Does it solve it to the expected extent? This is difficult to assess unless it has been done before (e. g., email system).

The needs, vision, goals, and objectives are the starting points for validating the system. It sets the "stake" in the ground and says this is what we want, what problem we intend to solve and to what extent we want to address the issues. (Performance metrics)

The *first part of validation* is to make sure that the system development starts out on the right track. This is done by VALIDATING the requirements of the system. Are these the "right" requirements that we will build to? Since this question needs to be addressed early in the project timeline, it requires high stakeholder involvement and an accurate translation of the needs, goals, and objectives into a set of system requirements that can be built. The system owner should take ample time to clarify the vision, goals, objectives and needs and to make them measurable. The translation of the needs into system requirements is done using the elicitation process and other techniques for example, looking at similar systems, technology review, prototyping, and modeling. *The second part of validation* is at the end of development where the system has been accepted and is now put into operations. Does the system do what it was intended to do and to what extent? Or otherwise was the "right" system built?

Verification is the process that makes sure that what was built matches the requirements for it. Was the system built the way the requirements and design specified? Or, was the system built "right"? Both the verification and validation processes are important and necessary, but it is the validation that views the system from the system owner and stakeholder perspective and it is the verification that the system is viewed from the development team's perspective. *Systems engineering's goal is to unify these views*.

4.6.2 Operations and Maintenance

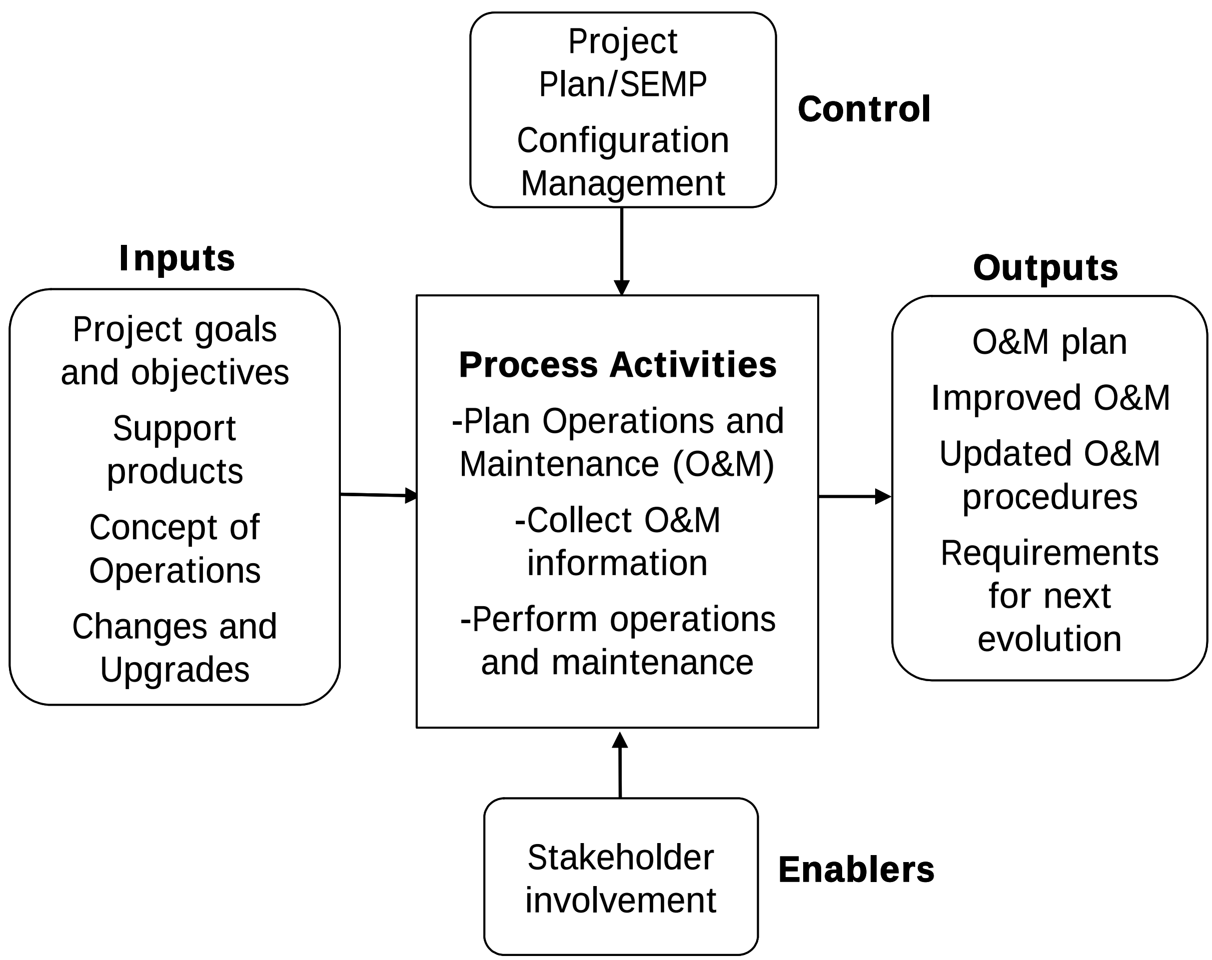
OBJECTIVE:

Effectively operate and maintain the system in a day-to-day operational environment.

DESCRIPTION:

Operations and maintenance involves planning for, and executing, activities such as: operating the system, monitoring system performance, making repairs, hiring and training operators, testing the system after any changes are made, and tuning the system. All systems require regular maintenance. Preventive maintenance involves inspection and proactive actions such as cleaning, replacement of components prior to the end of their rated life, backing up software and stored data, and replacing components that have become obsolete and unsupported. Reactive maintenance involves correcting faults when they occur. Software maintenance involves both correcting malfunctions (bugs) when they are discovered, upgrading components that become obsolete and unsupported, and making minor modifications as needed to improve functionality.

CONTEXT OF PROCESS:



OPERATIONS AND MAINTENANCE PROCESS

Inputs:

Project goals and objectives were identified in project planning

Support products such as users' manuals and maintenance guides were obtained during system development.

Concept of Operations describes the operational scenarios for which procedures are needed.

Changes and upgrades provide opportunities to enhance system operation and maintenance.

Control:

Project Plan/Systems Engineering Management Plan (SEMP) defines the overall operations and maintenance plan for the project including the goals and objectives.

Configuration management will be used to manage the synchronization of any changes that might occur during the maintenance of the system. This would include replacement elements (spare parts, units and subsystems) that would need to be documented as part of the physical audit of the system.

Enablers:

Stakeholder involvement is needed to ensure all parties have input and are aware of ongoing activities.

Outputs:

Operations and Maintenance Plan documents the procedures and resources, training and support needed for operating and maintaining the system.

Improved operation and maintenance will result for the life of the system.

Updated Operation and Maintenance Procedures will be developed as the system changes over time.

Requirements for next evolution are captured when identified by operations and maintenance personnel.

Process Activities:

Plan Operations and Maintenance

During the Concept of Operations phase of the project, two important views of the system are defined, the operations and maintenance views. These views, which envision how the system will operate and be maintained, become the initial planning for the system when it is commissioned into service. Once the system is commissioned into service, these plans are updated to reflect the as-is operational and maintenance environment. The complete Operations and Maintenance Plan should:

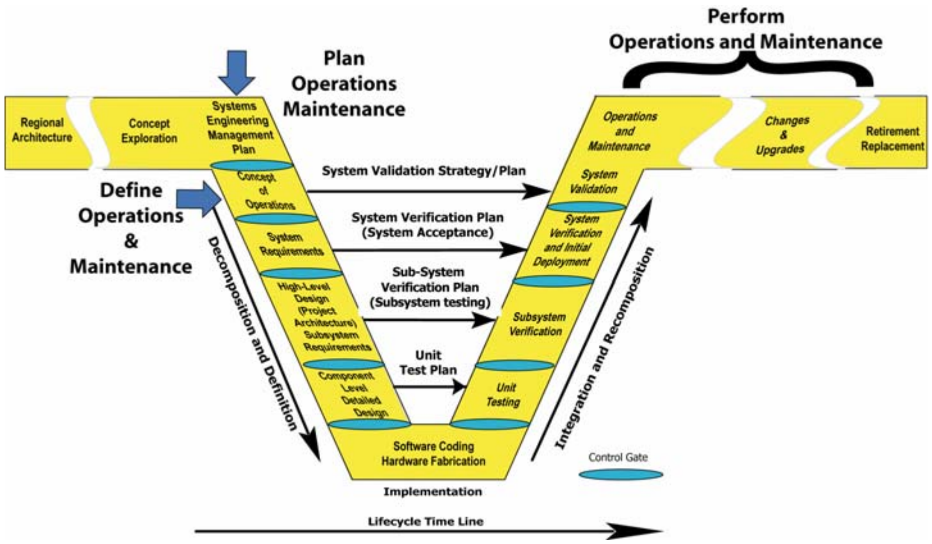
- Identify funding and policies supporting on-going operation and maintenance;
- Identify the aspects of the system needing operation or maintenance;
- Identify the manuals (users, administrators, and maintenance), configuration records, and procedures that are to be used in operation and maintenance.
- Identify the personnel who will be responsible for operations and maintenance.
- Identify initial and on-going personnel training procedures, special skills, tools and other resources.
- Identify operations and maintenance related data to be collected and how it is to be processed and reported.
- Identify methods to be used to monitor the effectiveness of operations and maintenance.

Collect Operations and Maintenance information

Operations and maintenance information should be collected throughout the operational life of the system including disruption in service of the system, restoration measures undertaken, and system performance. Down time and the mean time to repair should be documented and used to assess the average availability of the system. Repair logs should include vendor notice of obsolescence and notice of design changes that will affect the maintainability of the system elements.

Perform operations and maintenance

Operations and maintenance procedures need to proceed as defined in the Operations and Maintenance Plan. Over time the procedures will need to be refined and updated, either because the system changes or improved procedures are developed. The Operations and Maintenance Plan needs to be updated as well as the documented procedures, users' manuals, and maintenance manuals.



Where does Operations and Maintenance take place in the project timeline?

Is there a policy or standard that talks about operations and maintenance?

FHWA Final Rule (23 CFR 940.11) requires that the identification of procedures and resources necessary for operations and management of the system be determined in the systems engineering analysis for ITS projects funded with Federal money from the Highway Trust Fund, including the Mass Transit Account.

Which activities are critical for the system owner to do?

- Secure adequate funding and management support for on-going operations and maintenance.
- Identify and recruit appropriate agency stakeholders to participate in operations and maintenance.
- Review and approve the Operations and Maintenance Plan, including any updates.
- Arrange for on-going monitoring of system performance to ensure it is being operated and maintained adequately.

How do I fit this step to my project? (Tailoring)

Operations and maintenance are necessary for all systems of any size or complexity. After the ITS system is built, it is made operational and maintained in operational condition for as long as is needed. However, some systems, such as traffic signals, operate autonomously with little routine human input. They need only initial configuration and periodic review and fine-tuning of the settings. Others, such as a closed circuit television system, require hands-on involvement by a human operator as part of normal operation. But a traffic signal system may involve more intensive maintenance than a CCTV system.

The Operations and Maintenance Plan and associated documents such as manuals, operating procedures, and system configuration records, should record all the information needed for employees to keep the system operating effectively and for managers to plan for future resource needs. Information provided should include that needed for day-to-day activities, and also that needed to plan for occasional activities, such as periodic preventive maintenance and system upgrades. The Concept of Operations, System Requirements and design documents should be consulted as a checklist of all the system elements and operational aspects that may need coverage in operations and maintenance documentation.

What should I track to reduce project risk and to get what is expected? (Metrics)

During system development and implementation, there is no direct measure of the effectiveness of operations and maintenance planning. Once the system is operational, there are ways to monitor its on-going performance.

Although it is often difficult, attempt to measure the on-going operational effectiveness of the system, because this is a measure of the success of both operations and maintenance. If feasible, directly measure traveler experiences such as travel time and safety rates, either continuously or every year or so.

Otherwise, track indirect performance measures. Have operators record and periodically summarize notable operational successes and failures. Record maintenance actions in a way that enables calculation of statistics such as average number of failures per year and mean-time-between failures. Track the number of traveler complaints related to the system. Look for trends that suggest operation or failure rates are deteriorating. Look for ways to make the trend positive.

Are all the bases covered? (Checklist)

☑ Have you ensured funding and management support are in place for on-going operations and maintenance?

☑ Did you prepare and review an Operations and Maintenance Plan?

☑ Did you involve all stakeholders?

☑ Are resources and training in place for system start-up?

☑ Have you established procedures for continually monitoring the effectiveness of operations and maintenance?

☑ Do you have a plan for long term upgrades?

Are there any other recommendations that can help?

Stakeholders often underestimate or neglect the cost of operations and maintenance. Consider the cost of configuration management, as well as hands-on operation and maintenance activities.

Remember that most software requires maintenance. This is especially true of software operating on a general-purpose computer, but may also be true of embedded software in a specialty device. Even if no bugs surface, most software will need to be updated over time to adjust to changes in external interfaces, to upgrade or replace obsolete versions of third party components, to be moved to a new computing platform when the original one becomes unserviceable or inadequate, or to make minor modifications to the functionality to address new requirements, or needs that were overlooked during initial system development.

Configuration management (section 4.8.6) keeps the documentation synchronized with the functional and physical characteristics of the system.

Any information that may be needed in the future, for any aspect of operation, maintenance, retirement or replacement, should be recorded and kept up to date. It is not sufficient to rely on the memory of involved personnel for such information.

Beyond documentation, configuration management involves establishing and following rigorous procedures for controlling changes to the system. Change control ensures operations and maintenance personnel do not make inappropriate or undocumented changes to the system. A Control Change Board reviews and approves or rejects all proposed changes. A change can be as simple as changing a configuration setting to replacing a major system component. The Change Control Board includes representatives of all parties with an interest or involvement in the system, and ensures that all potential options and ramifications, including risk, are considered before proceeding. Development and implementation of any significant changes to the system should follow the same systems engineering process that was used for the original system development.

Use blanket approvals to cover routine maintenance. Routine maintenance procedures can be handled by blanket approvals of routine activities, but with regular review and a requirement to document all changes. Change control procedures should include periodic audits to confirm that procedures are being followed and that the functionality and physical characteristics of the system match those required by the approved configuration documentation.

4.6.3 Changes and Upgrades

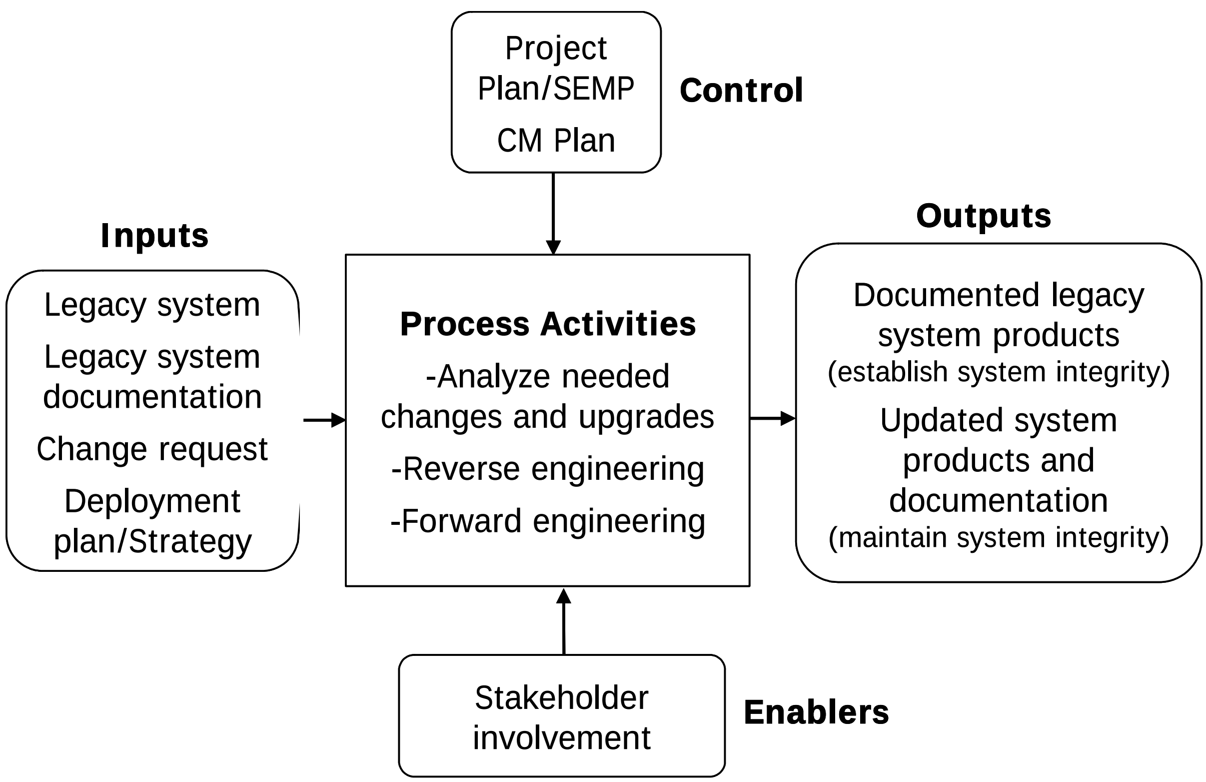
OBJECTIVE

This step allows the system owner to evolve the system to keep pace with changing needs, advancing and changing technology, and/or add system capabilities over time. These changes and upgrades will be performed in a systematic way to maintain or establish system integrity. ***Integrity*** in context of systems engineering means that the system's functional, performance, physical and enabling products are accurately documented by its requirements, design, and support specifications. The system documentation is accurate and sufficient to the point where changes and upgrades can be performed by any competent development team. This gives the system owner the freedom to have the widest possible selection of development teams for evolving the system.

DESCRIPTION:

The guidance in this step will address upgrades that are planned and ones that are based on new stakeholder needs. This step will also give guidance on implementing upgrades on a system that has not been well documented (s*ee integrity as defined in the objective above*). This step will also give guidance on handling commercial-off-the-shelf products and applications that may have become obsolete, changed in design, or beyond its contracted support life.

CONTEXT OF PROCESS:



CHANGES AND UPGRADES PROCESS

Inputs

Legacy System is the existing system to which the upgrade or change will be applied.

Legacy System documentation includes the requirements, design and support documentation.

Change Request identifies the new change or upgrade needs or requirements for the system.

Control

Systems Engineering Management Plan (SEMP) is used if the upgrade is a planned upgrade that was part of the development strategy.

Configuration Management Plan describes how planned and unplanned upgrades and changes would be evaluated, coordinated and inserted into the legacy system.

Enablers

Stakeholder involvement is important when the system is being changed or upgraded and is essential if the changes and upgrades will impact the stakeholders in some way.

Outputs

Documented legacy system products in the areas of change and upgrade. This is done if the legacy system has not been well documented before the change or upgrade.

Updated system products and documentation for the new capabilities as well as the impacted areas of the legacy system.

Process Activities:

Analyze needed changes and upgrades

Planned upgrades are done in accordance with the systems engineering management plan (SEMP). The SEMP may have a development strategy that lays out a plan for the evolution of the system over time. The plan may have several phases to the system evolution. For example, the first phase may deploy the communications network, phase 2 may deploy the CCTV (camera system), phase 3 the detection system and so on until the system has been fully implemented. Each of these phases should be implemented using the forward engineering process.

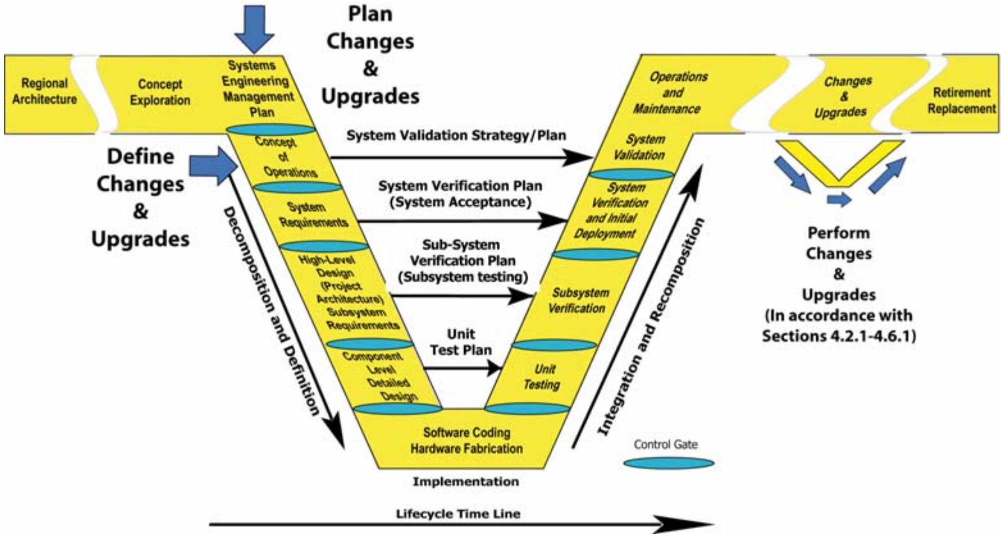
Unplanned changes may be the results of a change in needs, technology obsolescence, requirements or new stakeholder participation. If the system was well documented the changes should be implemented using the forward engineering process. The system owner’s configuration management process will lay out how the changes will be evaluated, coordinated and inserted into the system. If the system was not well documented then the reverse engineering process should be performed as described below. An analysis of the legacy system and its documentation is needed to assess to what extent, if any, a reverse engineering process is needed.

Reverse engineering

Reverse engineering is documenting the legacy system (the system being upgraded or changed). This includes the interfaces (both internal and external to the system), hardware, software, and support products (original development tools, test plans, and traceability matrix). This process requires one to analyze the system’s functionality, examine the software (source code), inspect the hardware, and then create or recreate a set of requirements and design documentation that matches the system as it currently exists. This is developing a set of “as-builds” for the legacy system in the reverse direction. This prepares the legacy system for the upgrades and changes.

Forward engineering

Forward engineering is the process of following the Vee Development Model as defined in sections 4.3-4.6 of this guidebook. All changes and upgrades to the system start with the update of the systems engineering management plan, concept of operations, followed by the requirements, sub-systems, high-level and detailed design. When evolving, upgrading or implementing changes to a legacy system, it should be in a forward engineering approach suggested in this guidebook.



Where do the Changes and Upgrades take place in the project timeline?

Is there a policy or standard that talks about Changes and Upgrades?

This task would include all the tasks from phase 0 to phase 4 including all FHWA Final Rule requirements and standards.

Which activities are critical for the system owner to do?

- Perform the critical activities for the system owner in each of the steps as described in sections 4.3-4.6.
- Elicit stakeholder involvement for the reviews of the products coming out of the reverse engineering process.
- Elicit stakeholder involvement in the workshops that are held with doing the reverse engineering process.

How do I fit these activities to my project? (Tailoring)

In the reverse engineering process, first identify the areas that are going to be impacted by the upgrades and changes. Those areas should be the focus of the reverse engineering activities. This will tailor the activity to only the affected areas of the legacy system. (**See Caution**)

In the forward engineering activities, apply the tailoring guides identified in sections 4.3-4.6.

What should I track in this process step to reduce project risks and get what is expected? (Metrics)

<u>*On the technical side:*</u>

- For reverse engineering, identify and track the extent of impact (e.g. number of software modules that need to change, number of interfaces that need to change) to the legacy system that changes and upgrades will have. This will help in estimating the effort to implement the changes.
- For forward engineering activities all of the technical metrics identified in sections 4.3-4.6 are recommended for tracking.

<u>*On the project management side:*</u>

- Reverse engineering is a discovery and a documentation effort. Task order with milestones is a way to track progress for this type of project. For example, the task is to document the software architecture in 6 weeks. By week 2- document top level software structure. By week 4 – document interfaces between major software modules, and by week 6 – document next level software modules.
- For forward engineering activities, all of the project management metrics identified in sections 4.3-4.6 are recommended.

Are all the bases covered? (Checklist)

☑ Is there a change management process in place?

☑ Is the documentation for the legacy system available?

☑ Have the upgrades and changes been clearly identified?

☑ If this has been a planned upgrade (evolutionary development), have the systems engineering management plan, concept of operations, requirements, and test plans been reviewed and updated?

☑ If this is a new capability that is added to the system, have the systems engineering plan, concept of operations, requirements, and test plans been developed for this new capability?

- ☑ Does the upgrade or change impact the project architecture? If so, is the updated project architecture consistent with the regional architecture?
- ☑ Prior to applying changes and upgrades, have the impacted areas of the legacy system been documented to a level that the changes and upgrades can be applied using the forward engineering process? (as described in sections 4.3-4.6)

Are there any other recommendations that can help?

If reverse engineering on a legacy system is needed, it should be done only to the areas that will be affected by the changes and upgrades. On major systems it may be too costly to document the entire legacy system for minor upgrades and changes. The cost effective approach is to document as needed and over time, as changes are made, more of the system will be documented. Those areas that never get changed will not be documented.

Continue the reverse engineering process through the implementation of the changes. The reverse engineering process will document the obvious impacted areas of the legacy system that changes and upgrades will be applied to. But as the changes are applied to the affected areas, the implementer must check and continue the documentation effort since changes to a system may impact areas not intended for change or to affected areas in a subtle way that was not anticipated. It will be the implementer and test support that will most likely uncover these types of issues and they must be ready to identify and document these areas.

A closer look at reverse engineering and its application to commercial-off-the-shelf products and applications

Reverse Engineering *is simply documenting anexisting Intelligent Transportation Systems functional (what it does – requirements), physical (how it does it – design), and support (the way it was built and maintained* – enabling products) characteristics. Legacy system documentation may not have been complete, lost or over time it was not kept up to date. Traditionally, system owners would use the system to the end of its life and start over. Today there is a regional focus on multiple agency involvement, fast-paced changes in technology, and constrained budgets that are driving the system owners to evolving their systems and to have greater latitude in development team choices, whether this is done in-house or contracted or both. In these situations reverse engineering may be a good alternative to starting over.

Reverse engineering for commercial-off-the-shelf products (COTS) and applications focuses on interfaces and modularity. Examples of some common elements that exist are workstations and operating systems, databases, changeable message signs, cameras, communications, detection and traffic control systems. Custom developments focus on user interfaces, data structures, distribution and applications that analyze exchange and translate information. Both the forward and reverse engineering activities should focus on allowing these COTS products to be updated or change out as they become obsolete, have changes to design, or reach the end of their service life. Database management system (DBMS) is a good example of this. When the DBMS reaches the end of life, the system owner can choose to stay with the existing DBMS, that is now unsupported, or migrate to the latest version of the DBMS. If the system owner chooses to upgrade, the impact may affect the current operating system and computer hardware. This then would impact other applications like the user interface, drivers for the camera system, changeable message signs, and communications. By keeping the applications modular, and interfaces (both internal and external) well defined, the impacts of obsolescence can be minimized.

4.7 System Retirement / Replacement

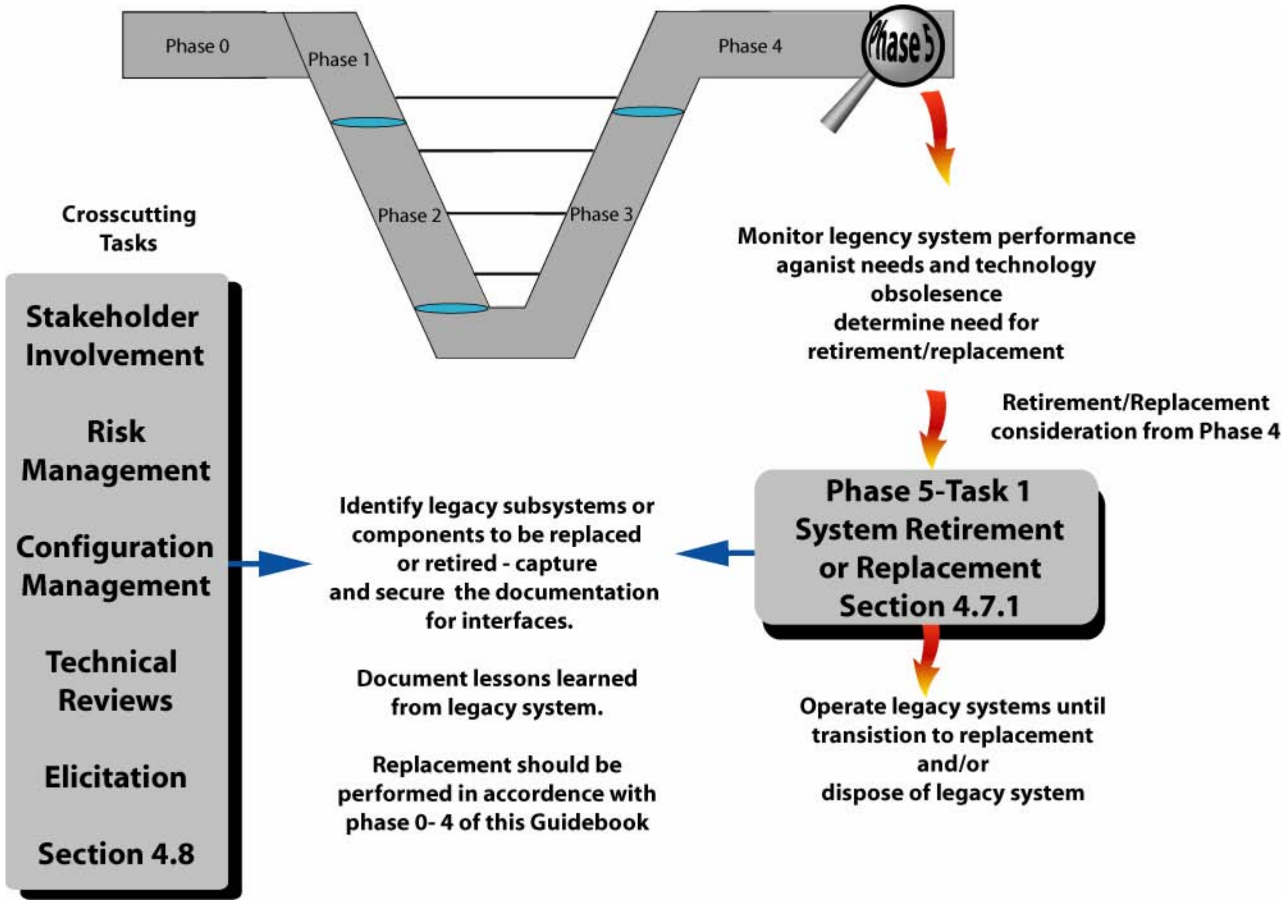


Figure 4-14 Phase 5 - System Retirement and/or Replacement Roadmap

4.7.1 System Retirement / Replacement

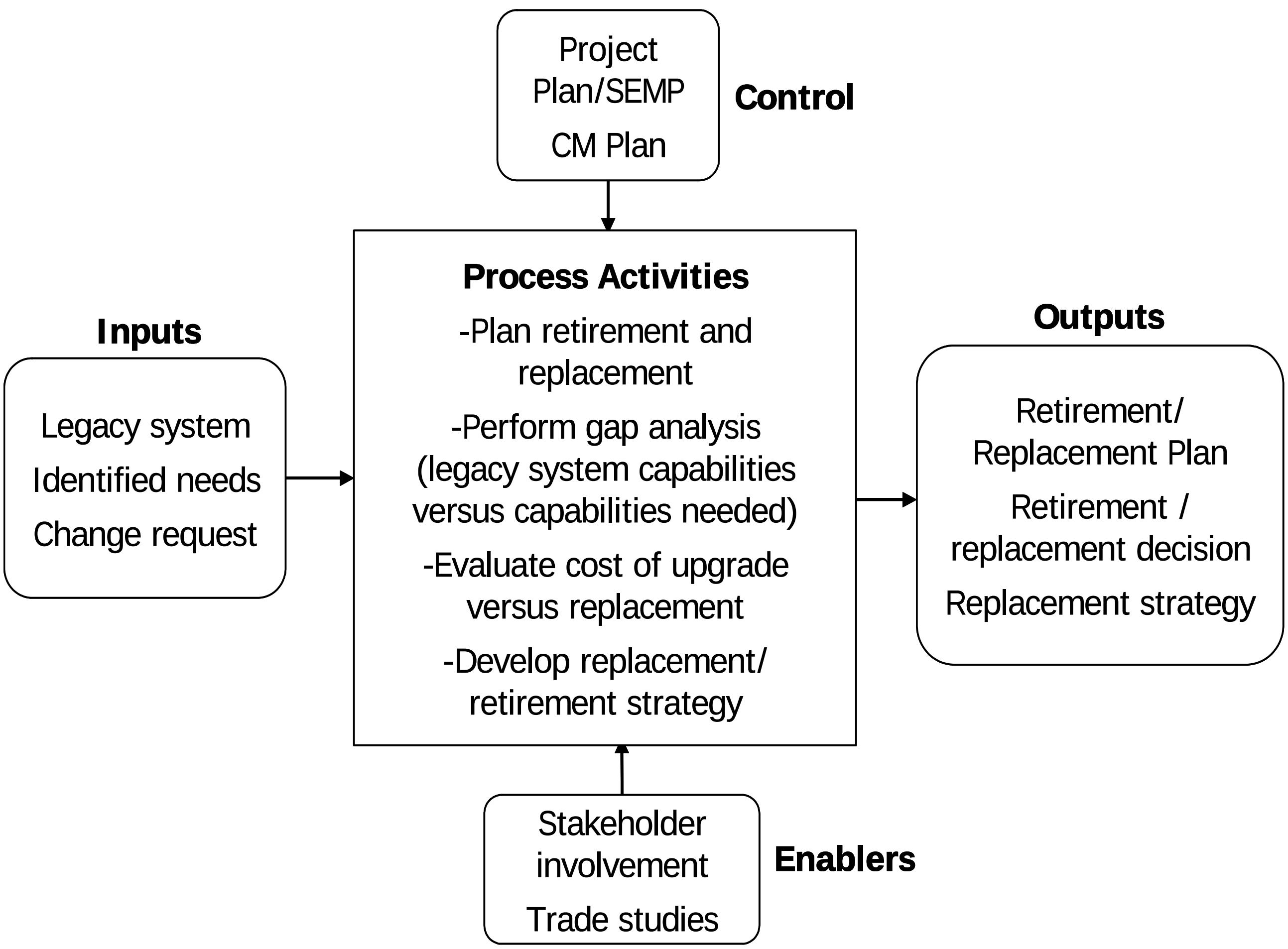
OBJECTIVE

To determine when an Intelligent Transportation System or major sub-system needs to be retired or replaced and provide guidance on a replacement strategy.

DESCRIPTION:

This step in the process provides guidance on determining the end of life for a system or sub-system. The end of life for a system or major sub-system can be a planned event or it can occur as a result of the following factors: high cost of operations and maintenance, capabilities of the system are no longer needed or cost effective, high cost of upgrades and changes, or technology obsolescence making the system/sub-system unsupportable. Eventually most system/sub-systems will face some major replacement no matter how well it was developed or maintained. To get the maximum useful life out of a system/sub-system, it must be well designed, documented, and maintained. The following are factors that will certainly shorten the useful life of a system or sub-system: lack of documentation, lack of an agreed to concept of operations, lack of adequate operations and maintenance budget, and a lack of an effective configuration management process that synchronizes changes with system documentation. When a system or sub-system needs to be replaced, a strategy on how the system or sub-system must be developed will become part of the systems engineering management plan for the next evolution of development.

CONTEXT OF PROCESS



SYSTEM RETIREMENT/REPLACEMENT PROCESS

Inputs

Legacy system is the system or major sub-system that is subject to retirement and or replacement.

Identified needs are the new needs that the legacy system is to address

Change request is the documentation that defines and describes the needed changes to the legacy system or sub-system to meet the new needs. This comes from the configuration management process.

Controls

Project Plan/Systems engineering management plan (SEMP), if this is a planned replacement or retirement, contains the development strategy that identified what would be replaced or retired, and the identification of the new capabilities and requirements.

Configuration Management Plan would have the processes documented that would evaluate the change history, costs, and impacts of changes

Enablers

Stakeholder involvement is essential. Stakeholders that will be affected by the change must be involved in the process of replacement or retirement of the system.

Trade Studies is the process tools that can be used to evaluate whether to replace or upgrade the legacy system. This enables the stakeholders to decide on the best cost effective approach.

Outputs

Retirement/Replacement Plan is part of the Project Plan/SEMP that provides the overall strategy for retirement and replacement of the system.

Retirement/replacement decision versus upgrading and changing the legacy system.

Replacement strategy documents the way the system or sub-system will be replaced. This will become part of the systems engineering management plan for the next evolution of development.

Process Activities:

Plan retirement and replacement

The initial planning of the project may include a replacement plan for the system or sub-system. This may be the case for the deployment of an interim system to address an immediate need. At the time of replacement the system owner and affected stakeholders should assess and review the plan to see that it is still viable. It is important to reassess the plan especially if needs have changed or there are different or new stakeholders involved.

Perform Gap Analysis: legacy system capabilities versus capabilities needed

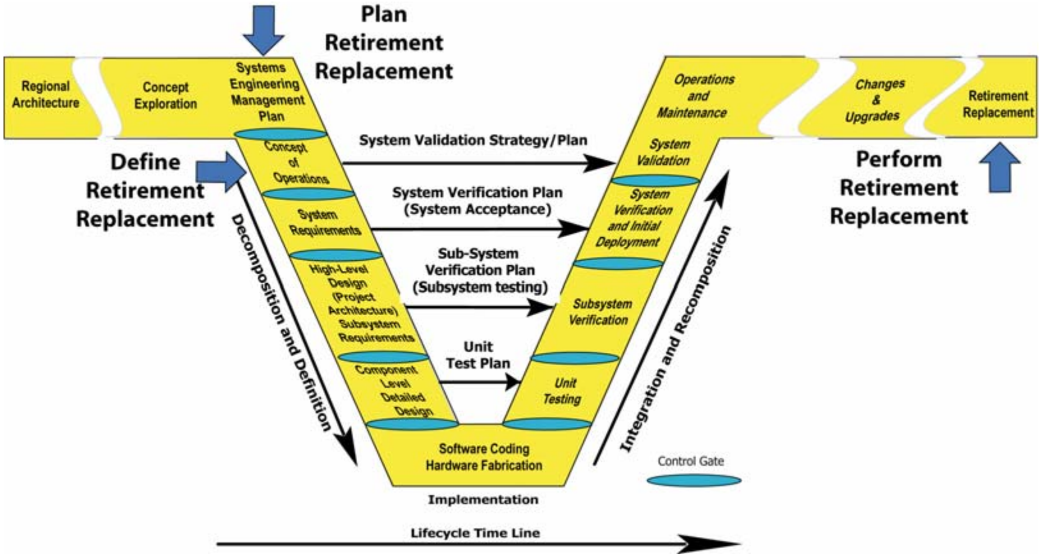
The trade studies process can evaluate the cost benefit of upgrading the current system or replacement of the entire system or some major sub-system(s). Can the current system evolve to meet the new needs? Is the technology used in the current system obsolete and no longer supportable? Are the operations and maintenance costs to the point where a replacement system is more cost effective?

Evaluate cost of upgrade versus replacement

The trade study should include lifecycle costs analysis, including the operations and maintenance costs as well as the replacement costs. Issues to address in the evaluation are the vendor support of COTS products and license costs. Is the cost of documenting the existing system prohibitively expensive?

Develop replacement/retirement strategy

A strategy on system or sub-system replacement needs to be planned for the replacement of Intelligent Transportation Systems. This planning may require the upgrade of facilities, floor space, air conditioning communications, furniture, and other such facilities. Some Intelligent Transportation Systems are safety critical and have to be operational full time. In this case the new system would need to be deployed in parallel with the legacy system and a switch over plan created that would allow the legacy system to act as a back up while the new system is being verified and validated. This has a cost and deployment impact on having the two systems fielded for a period of time. In other cases, functionality may not be safety critical and in these cases the removal and then deployment of the new system or sub-system may be more cost effective.



Where does the Retirement/Replacement activity take place in the project timeline?

Is there a policy or standard that talks about Retirement/Replacement?

This task would include all the tasks from phase 0 to phase 4 including all FHWA Final Rule requirements and standards.

Which activities are critical for the system owner to do?

- Participate in the reassessment of the replacement/retirement plan. If in the original development, this was a planned replacement or retirement, reevaluate the plan to see if the planned replacement is still needed.
- Be involved in the assessment of alternative replacement systems or sub-systems.
- Participate in the Configuration Management process to assess the cost of upgrade to the legacy system versus its replacement.
- Elicit stakeholder involvement and support for the upgrade or replacement decision.
- Participate in the development of the replacement strategy for the system/sub-system.

How do I fit these activities to my project? (Tailoring)

The replacement strategy can be tailored for the project but factors that constrain this will be if the legacy system or sub-system is critical to public safety and needs to be operational nearly full time. Are there alternates to the legacy system or sub-system operations that can allow it to be inoperable until the new system is in place, verified, validated and operational?

What should I track in this process step to reduce project risks and get what is expected? (Metrics)

<u>*On the technical side:*</u>

- In replacing a system or sub-system, identify the new and needed capabilities (functions) that will be added to the legacy systems/ sub-system capabilities. Will you, as the system owner or stakeholders, be gaining or losing needed capabilities over what you currently have?
- Track and manage the technical documentation of the new system/sub-system. Is the new system/sub-system well documented? For example is the following documentation available to the system owner and stakeholders:
- Track requirements, design, interfaces, support products (e.g. verification, maintenance, production, development and training) documentation

<u>*On the project management side:*</u>

- Identify the lifecycle cost of the new system/sub-system. Will the new system/sub-system have an improved cost/benefit ratio in operations and maintenance cost over its life? (The new system should work better and cost less to maintain.)

Are all the bases covered? (Checklist)

☑ Was a trade study done on the cost/benefit of upgrading the legacy system/sub-system against the cost/benefit of developing or procuring a new system/sub-system?

- ☑ Did the trade studies include the operations and maintenance costs of the legacy and new systems/sub-systems?
- ☑ If there was an initial plan to replace a system or sub-system, has that plan been reviewed prior to replacement to assess if it is still viable?
- ☑ Is the new system/sub-system well documented? Does it have at a minimum:
  - ☑ New concept of operations
  - ☑ Requirements documentation
  - ☑ High-level design documentation
  - ☑ Detailed design documentation
  - ☑ Verification plans
  - ☑ Support documentation on development, training, maintenance, and users manuals.
- ☑ Is there a replacement strategy to switch out the legacy system or sub-system with the new system or sub-system?
- ☑ Have all of the affected stakeholders been involved in the replacement/retirement decision and the planning and replacement strategy for the new system/sub-system?

Are there any other recommendations that can help?

TIP

When a system needs to be replaced, do it in an incremental manner (sub-system by sub-system). Here are examples of replacement strategies for two different types of systems, A <u>Traffic Control System</u> and a <u>Regional Advanced Transportation Management System</u>.

<u>Strategy for replacing a Traffic Control System</u>

Option 1 Deploy the new traffic control system in parallel with the legacy traffic control system and incrementally add intersections and replace or reconfigure the field controllers based on the current segmentation of communications layout; running both systems at the same time.

Option 2 One might run "time of day" as the field controllers remove the legacy central host, and deploy the new traffic management system central host and add intersections incrementally replacing or reconfiguring the field controllers. The strategy would depend on the current availability, flexibility and accessibility of the communications infrastructure that exists.

<u>Replacement of the host software in a Regional Advanced Transportation Management System (ATMS)</u>

In this situation, the replacement strategy is largely driven by the project architecture of the legacy system and the modularity of the legacy software.

If well defined interfaces exist between the field devices and the ATMS host, the replacement strategy is done at these interfaces. The new host system is deployed in parallel with the legacy system and an incremental switch over is made sub-system by sub-system. For example, the changeable message sign sub-system is switched over, then the camera sub-system, then the ramp metering sub-system, then the detection and incident management system. Stand alone functions are the easiest to switch over but integrated functions such as the detection and incident management functions will be more difficult. The important issue here is to be able to switch back if needed. If the software of the legacy system is such that removing a sub-system causes the legacy system to act in an unpredictable way, temporary software or hardware simulators may be needed to simulate the missing sub-system from the legacy system until the switch over is completed. The switch over will require additional staff since two systems are running in parallel.

The overriding concern is the safety to the public. These switch-over events should be done on off-peak hours such as middle of the night or weekends, or diverting traffic to a safer route until the switch-over is completed and tested.

If the interfaces to the field devices are not well defined, it is recommended that the interfaces to the field devices be developed first before adding the host system.

4.8 Crosscutting Activities

This section identifies the needed activities that support the systems engineering process steps identified in the previous sections. Each of these crosscutting activities supports one or more of the process steps and in most cases are shown as Enablers and/or Controls. These crosscutting activities are processes and support each other as well as the systems engineering process steps.

The following is the list of crosscutting activities that have been identified as part of this Guidebook.

- Stakeholder Involvement
- Elicitation
- Project Management Practices
- Risk Management
- Program Metrics
- Configuration Management/Interface Management
- Process Improvement
- Control Gates
- Trade Studies
- Technical Reviews

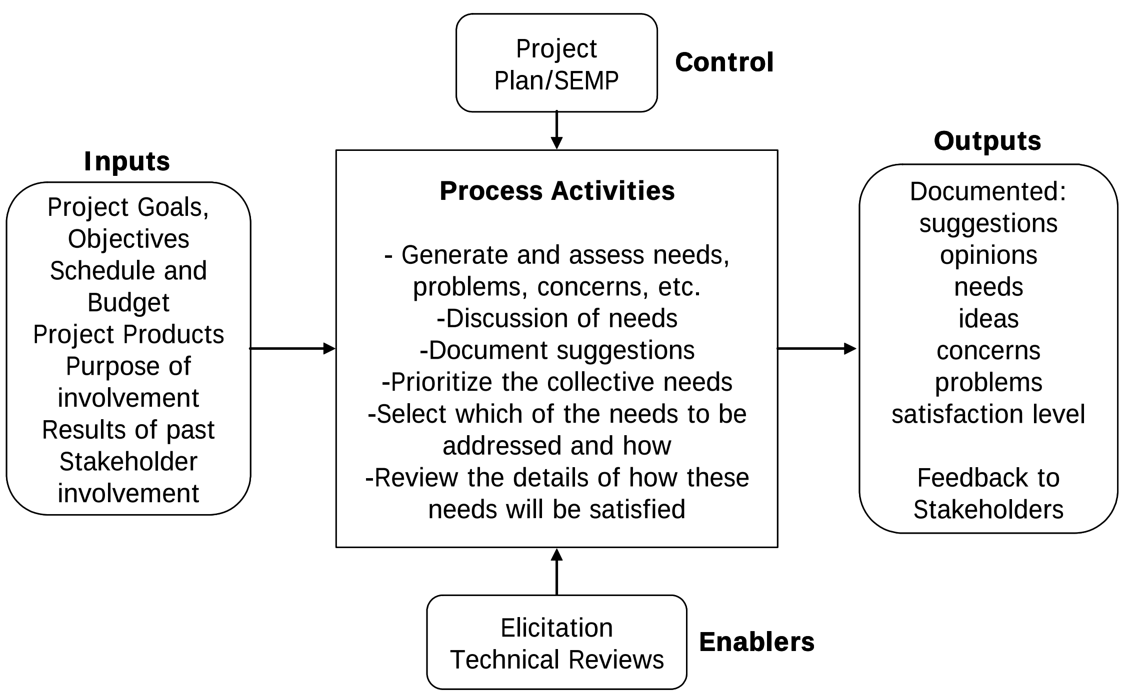
These activities are critical to the success of development of Intelligent Transportation Systems and, in the case of configuration management, extend throughout the life of the system. The crosscutting activities provide a set of industry best practices that support the gathering of information, provide checks and balances needed to ensure the quality of the product, and facilitate proactively planning, monitoring and controlling each of the systems engineering process steps.

4.8.1 Stakeholder Involvement

OBJECTIVE:
Stakeholder involvement insures that needs, problems, issues, constraints and priorities are examined, prioritized and addressed during each stage of the project's development. Rarely if ever does a single project satisfy every need of every stakeholder. But the team can and should understand what all the most important needs are, and can thus make sure that the project satisfies the most important of these within the project's time, and budget. The team can, and should, make conscious, well thought-out, well discussed, and well reviewed decisions as to what all the stakeholders' important needs are, and as to which are going to be satisfied, which are not, and why these decisions are being made. To do this requires that the stakeholders themselves participate heavily in the earliest phases, and continue to participate throughout as the project evolves.

DESCRIPTION:
Stakeholders are all the agencies, groups and individuals who will be affected by the system. Stakeholders include planners, users, and system owner agencies may be the primary operators and maintainers of the system, partner agencies who will utilize the system. Sometimes stakeholders include the public or portions of the public. Each stakeholder can bring a wealth of experience, wisdom, knowledge and insight from the perspective of their own organization or stakeholder group. They also bring needs and issues that need to be addressed. A representative from each stakeholder group should be included as participants in the project. For instance, there might be a representative from each local, state and federal agency, and from emergency services. Some projects are very diverse and require participation from many agencies and users. Other projects have fewer stakeholder groups. The person who represents each stakeholder should be fully aware of the stakeholder-group's history, problems, and current needs. They should be familiar with what has worked in the past, and what has not worked. And they should be a valid representative of their stakeholders group, in other words, they should accurately reflect their needs and expectations. Each of the chosen representatives should be consulted frequently, one-on-one and in groups involving other stakeholders. Their opinions and suggestions should be encouraged, inspected, and given respectful consideration.

CONTEXT OF PROCESS:



STAKEHOLDER INVOLVEMENT PROCESS

Inputs:

Project Goals, Objectives, Schedule and Budget (most recent version) provide an understanding of the environment of the project and the limits of time and money the project has remaining to work with.

Project's major outputs to date (the most recent version) provides a view of what has already been decided.

Purpose of stakeholder involvement orients the stakeholder as to what the purpose of this particular session is, and what the overall purpose of stakeholder involvement is.

Results of past stakeholder involvement enables them to see the effects of previous stakeholder input, review what has been addressed, and see how their efforts are helping both the stakeholder group and the project.

Control:

Project Plan / SEMP defines the tasks, schedule and processes to be employed for involving stakeholders.

Enablers:

Elicitation provides techniques for gathering stakeholder input and engendering constructive brainstorming, discussion, understanding, suggestions, and ideas.

Technical Reviews provide a formalized setting for enabling stakeholders to see what the outcomes of their inputs have been so far, and for ensuring them that their most important concerns have been addressed.

Outputs:

Documentation of stakeholders suggestions, opinions, needs, ideas, concerns, problems, satisfaction level includes recording all stakeholders ideas voiced during the session, re-writing it to make it clear and easy to understand, adding summary diagrams, lists and text, and describing how it affects the project.

Feedback tells stakeholders and other project staff what new information and insight was revealed.

Process Activities: *(Have stakeholders participate in)*

Generating and assessing lists of needs, problems,and concerns

Candidate lists of needs, problems, concerns, issues, and constraints are developed first by the core team of the project. These lists are then reviewed by stakeholders, first one-on-one, and later in a group, so that stakeholders can add any missing items that are important to the groups they represent. Then have each stakeholder make an assessment of a number of characteristics (e.g. cost, risk, utility, importance) for each item on the list, from the stakeholder group's point of view.

Discussing the needs of all the stakeholders with all the other stakeholders

This will enable each stakeholder to see these items as perceived by other stakeholders. It will enable the group to appreciate the needs and problems of other stakeholders, to understand where there are both synergistic and conflicting needs/solutions between different groups, and break down institutional barriers.

Making suggestions on how the most vital needs of all the stakeholders can be most cost-effectively satisfied

This will enable each stakeholder to benefit from the wisdom and experience of other stakeholders to help resolve conflicts and suggest solutions that can bring the greatest benefit to all the stakeholders as a group.

Prioritizing the collective needs of all the stakeholders

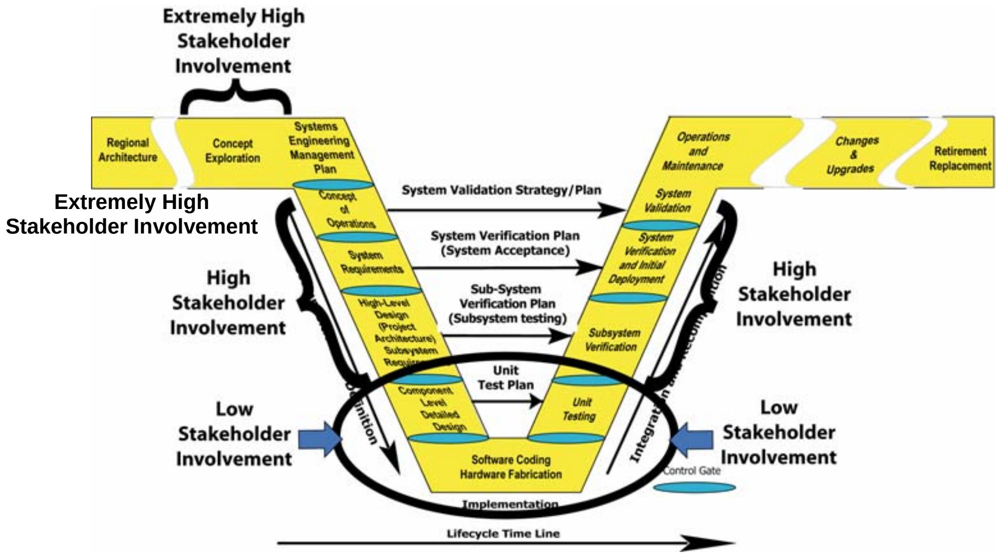
This will enable stakeholders to have a say in the prioritization process, to observe and influence what is selected, and to ensure that the stakeholder group is adequately represented during the process. It will also enable them to tell the rest of the stakeholder group why items were assessed, and prioritized as they were.

Selecting which needs will be addressed and how they will be addressed

This will enable stakeholders to highly influence the selection process, and to understand thoroughly how and why the project solutions evolved.

Reviewing the details of how these needs will be satisfied at each stage of the project

As the project evolves, stakeholders should review how the stakeholders' needs and problems are being addressed so that they can help the project team abort any faulty solutions, mitigate risks, fine-tune the solutions and improve the utility and cost-effectiveness of the system.



Where does Stakeholder Involvement take place in the project timeline?

Is there a policy or standard that talks about Stakeholder Involvement?

FHWA Final Rule (23 CFR 940.11) requires identifying the roles and responsibilities of participating agencies and stakeholders in the operation and implementation for ITS projects funded with Federal money from the Highway Trust Fund, including the Mass Transit Account.

Which activities are critical for the system owner to do?

- Identifying what the stakeholder groups are
- Getting an appropriate person or people to represent each important stakeholder group (These "stakeholder representatives" then become part of your development team, and participate in your stakeholder involvements sessions)
- Ensuring that stakeholder ideas, opinions, needs and concerns are used to decide what needs the system will address, and how the system will address them to ensure that the resulting product gives the highest benefit to the stakeholders for the time and budget allowed.

How do I fit these activities to my project? (Tailoring)

Some projects naturally involve many more stakeholder groups than others. The more stakeholder groups you have, the more stakeholder-group involvement sessions you will need in order to build consensus.

Some projects are quite similar to previous projects. Other projects are not similar to anything that has been done before. In general, the higher the similarity to previous successful projects, the less time and scrutiny you will need from your stakeholders. Conversely, the more your intended system differs from anything previously done, the more input you will need from your stakeholders.

What should I track in this process step to reduce project risks and get what is expected? (Metrics)

On the project management side:

- % of stakeholder involvement activities that occurred on time and within budget (as compared with the project plan)
- % of stakeholder groups represented in each stakeholder-involvement activity
- Level of satisfaction of each stakeholder group (as reported by its representative) with the project decisions, plans and processes to date
- For each stakeholder group, % of the most critical needs, problems, issues, and concerns addressed
- Level of satisfaction of each stakeholder group that their most important needs, problems, concerns and issues have been adequately addressed at each point in the project

Are all the bases covered? (Checklist)

☑ Have you discussed the projects goals, objectives, schedule and budget with each stakeholder rep (representative), and answered any questions they have?

☑ Have you informed each stakeholder rep of why stakeholder involvement is critical to the success of the project?

☑ Have you told each stakeholder rep what all the other stakeholder groups are that you are planning to involve, and why you're involving each?

☑ Have you demonstrated to each stakeholder rep how their participation will benefit the stakeholder group they represent?

☑ Have you explained to each stakeholder rep how past stakeholder participation has affected the project – how it has improved it, changed it, – and what the results were of past involvement?

☑ Have you described to each stakeholder rep what you want from them (both now and in the long-term)?

☑ Have you asked for each stakeholder rep's feedback, and written it down, being careful to note all of their ideas, concerns, needs, problems as they relate to the project?

☑ Have you utilized all of the stakeholder reps' feedback in developing and prioritizing the needs, concerns, issues, and alternative solutions?

☑ Have you utilized all of the stakeholder reps' feedback at each point in the project's development?

☑ At all points in the process, have you answered all the stakeholder rep's questions, and queried them on whatever you don't understand about their needs, problems, critical issues?

☑ Have you assessed each stakeholder rep's satisfaction level with the project processes, plans and decisions to date?

☑ Have you provided each stakeholder rep with feedback on the results of the stakeholder-involvement activities?

☑ Have you assessed how satisfied each stakeholder rep is that their needs, concerns have been adequately addressed?

☑ Have you expressed appreciation for each stakeholder rep's time, energy and ideas after each stakeholder involvement session?

Are there any other recommendations that can help?

A Closer look at stakeholders

There are often many levels of Stakeholders.

Primary stakeholders are those who have the biggest stake in project, usually those who will be operating, using, maintaining and/or funding the system. For example, a Traffic Information System (a system which collects and provides information on traffic conditions, accidents, alternative routes, weather and road conditions that affect traffic), primary stakeholders would include each department of transportation (perhaps both state and local) that collects and/or uses this information to help improve safety and traffic flow. Other primary stakeholders would include the police and emergency services who use and provide information to the system. If there are private groups (such as Information Service Providers) who collect and disseminate part of this information, they too are stakeholders.

There are also segments of the public who are stakeholders. They may include commuters, the handicapped and elderly, and commercial vehicle organizations. ***A given project may or may not have such segments of the public represented by a specific person.*** It may simply remember to explicitly identify and include the interests and needs of such users. Sometimes surveys are used to assess the needs, problems, concerns, and issues of such segments of the public. Sometimes organizations who service these segments of the public are queried. For instance, drivers of vehicles that transport the handicapped or the elderly may be queried. For example, the Automobile Club (AAA) might be contacted to provide information on typical needs of the traveling motorists they service.

Some projects may have as many as 20 or 30 stakeholder groups represented. (More than this number becomes unwieldy to use in a discussion group or workshops). Some projects may have as few as 3 to 5 stakeholder groups.

A closer look at the role the operating organization stakeholder is expected to perform.

It is these eventual operators who have the most knowledge of the environment in which your system will operate and have, or soon will have, the best opinions on how well your system will help them to do their job. Your own understanding of the operating domain is your first resource in designing the system. However, their deeper and more extensive understanding of the operating domain, tempered by their possibly limited understanding of the potential of your system, is a second resource that must be used to validate your Concept of Operations and to develop the requirements of your system.

Normally, when above-described stakeholder involvement process is used conscientiously and thoroughly, *the stakeholders naturally develop a sense of ownership and pride in the evolving system. In fact, many of the stakeholders will eventually become champions of the system.*

When a system truly helps the stakeholders with their most pressing problems and needs, stakeholders naturally will champion the system. The only way to make sure that your system truly meets the most important needs of all the high-priority stakeholders is to have them provide you with the immense experience and knowledge base they each have, and to tap into their collective expertise and insight into how to solve their common, and sometimes conflicting, needs.

When soliciting feedback from any stakeholder, whether it be regarding their needs, concerns, issues or anything else, it is best to take a cut at it yourself first, and use your "starter cut" as a strawman that they can modify.

Many people draw a blank when simply asked "What are your needs?" Or "what are your top priorities?" However, if you tell them that you are there to solicit their input in making up a list that adequately includes their stakeholder group's opinions, and then show them a strawman list, they are very likely to be able to give their opinion on how it should be changed.

It is important to provide the leadership and vision that draws your stakeholders into participating and taking an interest in your project.

Good leadership includes imparting your vision of the project – why it is needed, how it will help solve current problems, how it will benefit each of the stakeholder groups. Show your stakeholders that you are interested in the needs, issues, problems, and suggestion of every stakeholder group, and that this group of stakeholder representatives is vital to finding the greater good for the collection of stakeholder groups. Tell them you need their input. Ask them for their input. Give due respect to every piece of input and every suggestion they make. Encourage them to respect each other's needs and problems. Be a good moderator, giving everyone a chance to express their opinion, and aborting petty side arguments and bickering. Make suggestions yourself. Ask other for feedback on various suggestions. Brainstorm with your stakeholders. Empathize with them. Show them gratitude for their inputs, even if many issues remain unresolved. Keep the group on track, seeking solutions for the project. Keep the group informed on how their past participation has helped the project, and on what their future participation will be. Provide them with survey and questionnaire results. Keep the discussions positive, and abort any destructive activities (blaming, shaming, put-downs, or insults.) if they occur. These actions will help achieve convergence (vs. divergence) of ideas and concepts. It will also help break down institutional barriers and aid stakeholders to work towards the greater good.

Keep the interactions with your stakeholders regular, predictable, and ongoing throughout the project.

The initial contacts with stakeholders may be via one-on-one sessions, where you explain your project vision to them, and help them identify a person appropriate to see that their agency's needs and issues are adequately addressed. Ensuring contacts should include sessions where all the stakeholder representatives interact with each other while you set the agenda and moderate these sessions. When you set the program schedule, include such stakeholder sessions (e.g. workshops, or reviews) at regular and pre-scheduled intervals. You will need much more such interactive sessions early in the project to make sure you have fleshed out all their important needs, issues, problems and concerns, and have them help you to prioritize these needs. You will need to make use of surveys and questionnaires, and to provide feedback on the results of these surveys and questionnaires. You will need their help in identifying alternative candidate solutions, and in pointing out the pros and cons of each alternative. Once the initial set of needs and alternatives has been clearly identified, discussed, and evaluated, you will need to continue to provide feedback to the group on how these are being used to flesh out the details on the evolving system. It is critical that they review all your major decisions, prioritizations, evolving designs of the system and its interfaces, and to point out what elements they feel satisfactory, and where improvements are needed.

4.8.2 Elicitation

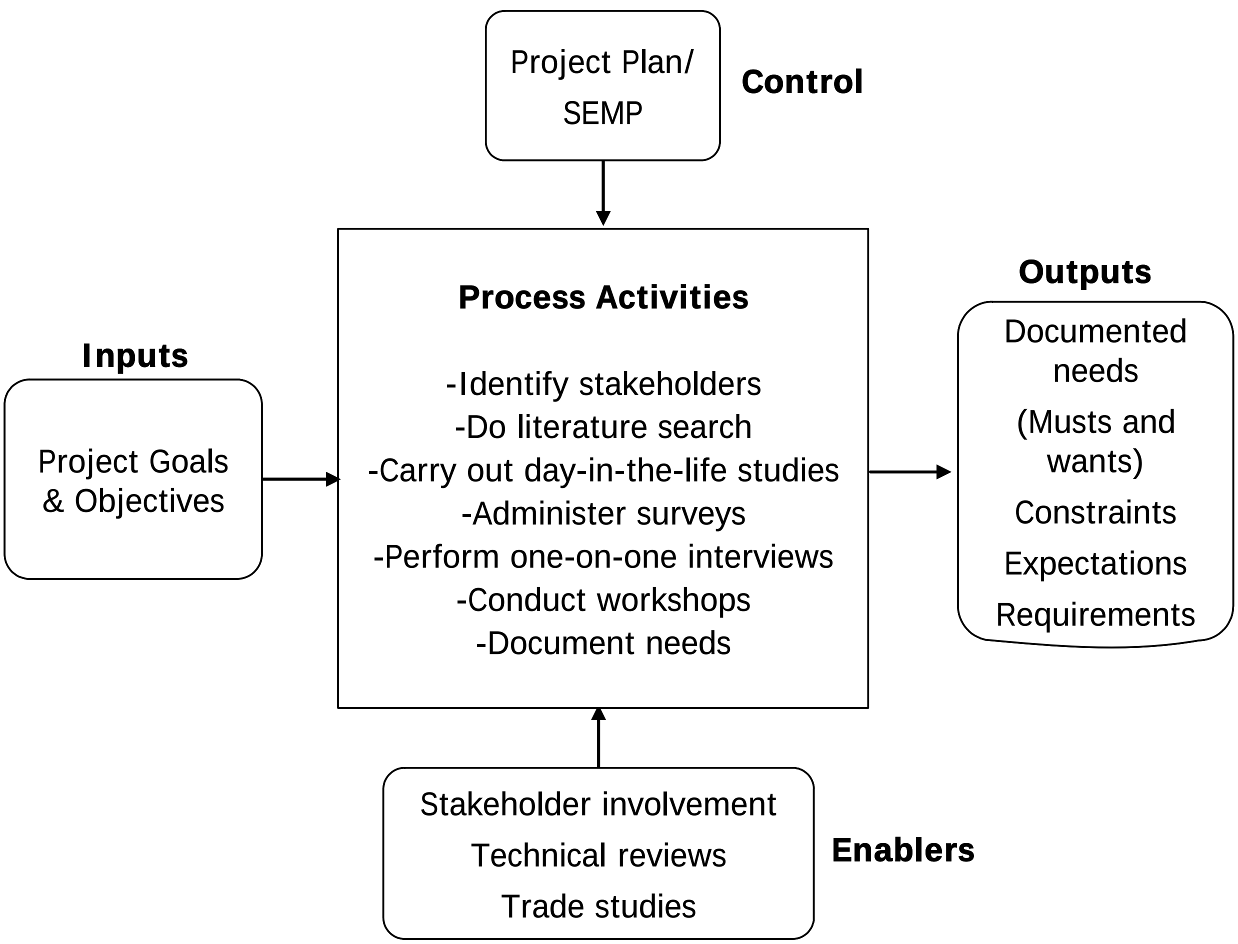
OBJECTIVE:

Elicitation is a set of techniques for drawing out stakeholder needs, goals, requirements, constraints, priorities, normal operations and preferences. It is done early in system development to support the initial needs assessment and hence development of the requirements. As the project progresses, the process is revisited as necessary to provide further clarification.

DESCRIPTION:

Elicitation is a collection of techniques to draw out and clarify stakeholder needs and requirements. Multiple techniques are provided to address the needs from various directions. Initial needs are usually vague, implicit (unstated), or described in terms of technical solutions. These techniques help the stakeholders to clarify their needs. The techniques present a logical sequence, starting with available material, and building on what has been learned through additional feedback. The actual steps taken for a project depend on the size and complexity of the project and the number and diversity of the stakeholders.

CONTEXT OF PROCESS:



ELICITATION PROCESS

Inputs:

Project Goals and Objectives are the major drivers for defining the needs.

Control:

Project Plan, SEMP will describe the elicitation approach, which will be developed before elicitation begins.

Enablers:

Stakeholder involvement is essential to defining valid and meaningful needs.

Technical reviews are an effective means to get stakeholder feedback on the needs being collected.

Trade studies support prioritization of the needs.

Outputs:

Key needs are the documented list of collected needs, their sources, and rationale for the selection of the key needs.

Constraints as well as needs are collected during the elicitation process. They are anything expressed by the stakeholders that may limit solutions to the needs.

Process Activities:

Identify stakeholders

Identify the stakeholders who will operate, maintain, use, benefit from, or otherwise be affected by the system. See 4.8.1 for details.

Do literature search

Take advantage of any existing documents, such as previous studies, reports, standards, specifications, scopes of work, or concepts of operations. Do your homework before you meet with stakeholders. Build on what you learn to make the other activities much more effective and focused.

Carry out day-in-the-life studies

The purpose is to understand current operations from the view of the key stakeholders. This is especially useful with system operators. Spend time with the stakeholders and document what they do and how they do it. Identify and document workflow threads; these will be the basis for scenarios in the Concept of Operations. Ask them what they like and do not like about how they currently do their job.

Administer surveys

Surveys are especially useful in setting priorities among multiple stakeholders or when there is insufficient funding to meet all of the important needs. First decide exactly what you hope to learn from the survey. Get expert assistance to design the survey carefully, asking questions in multiple ways and from both positive and negative views to prevent biasing the results and to clarify the answers.

Perform one-on-one interviews

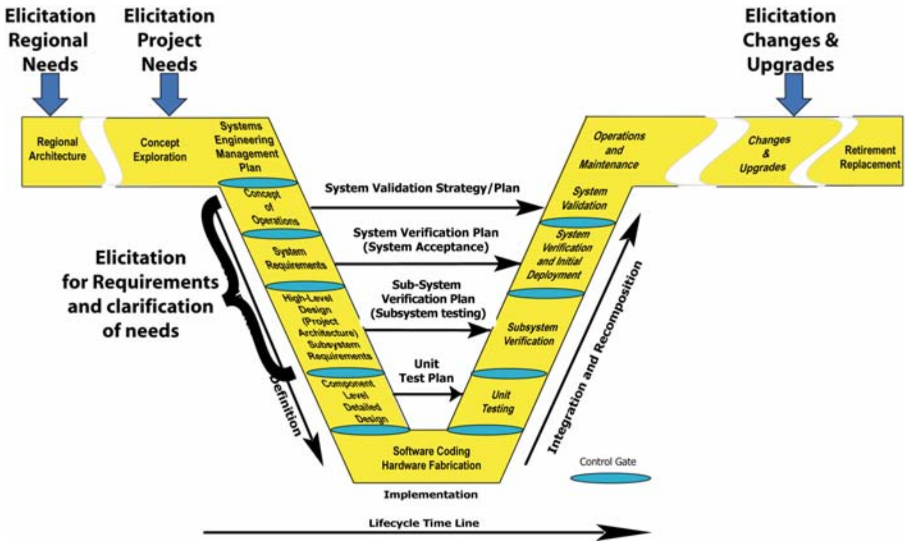
This is an opportunity to probe deeper into the perspective and needs of the individual stakeholders. Focus on the expertise of domain experts, but be especially aware of hot buttons and conflicting goals.

Conduct workshops

Workshops are an opportunity to “de-conflict” needs and requirements. Present the stakeholders with a summary of what you have heard so far, and describe the issues. Create a positive environment (a professional facilitator may help) in which the various groups can listen to each other’s concerns. Facilitate discussion and consensus.

Document needs

Document what has been learned in the elicitation process. Review it with the stakeholders and revise, as necessary.



Where does elicitation take place in the project timeline?

Is there a policy or standard that talks about Elicitation?

FHWA Final Rule does not specifically mention general elicitation practices to be followed. CMMI provides some useful material in this area.

Which activities are critical for the system owner to do?

- Identify stakeholders and encourage their participation.
- Participate as stakeholders in elicitation activities.
- Review summaries and conclusions of the elicitation process.

How do I fit these activities to my project? (Tailoring)

All projects require an identification of the stakeholders and documentation and acceptance of the findings. Beyond that, the combination of techniques used depends on the complexity of the system under development. A small, straight-forward system may only require a literature search. This is especially true if the needs have been well thought out and described in a document. But even in that case, an informal one-on-one interview is helpful to clarify the document.

A day-in-the-life study is important when the system will change operations, or if it is being developed to enhance operations. Surveys are needed to set priorities in systems with vague or contentious needs or an insufficient budget. One-on-one interviews are always recommended, but expand with the complexity of the project. For example, if new detector technology is to be installed, it is useful to talk to experts in that technology and, if possible, people at other agencies who have used the technology. Workshops may be as simple as a presentation and acceptance of conclusions, but they are essential when there are multiple agencies involved, especially if they have not worked together previously.

What should I track in this process step to reduce project risks and get what is expected? (Metrics)

On the Project management side:

- Percentage of relevant documents that have been utilized
- Percentage of stakeholder groups/individuals who have been queried using at least one of the techniques.
- Number of stakeholders who have agreed to the conclusions of the elicitation process

Are all the bases covered? (Checklist)

☑ Have all relevant stakeholders been identified?

☑ Have all appropriate techniques been used to draw out needs and requirements?

☑ Have all implicit needs been uncovered?

☑ Have all stakeholders agreed with the conclusions?

Are there any other recommendations that can help?

You may find that a *professional facilitator* helps to elicit needs, especially if there are conflicting needs. Also there are techniques for collecting, analyzing, and prioritizing needs that they can bring to bear.

Any collected needs must be tempered by reality. As you collect needs be aware of potential cost overruns, risks, conflicts, or scope creep. Here are some metrics to keep in mind and in front of the stakeholders.

- Estimated cost of meeting the expressed needs or requirements.
- Estimated risk level of the expressed needs or requirements.
- Number of expressed needs that conflict with those expressed by other stakeholders.
- Number of new requirements that are beyond the initial needs statement, since they signal a risk of scope creep.

Do not accept *stated needs* at face value without some exploration. Initial needs are often expressed in terms of solutions. For example, a need for more loops is really a need for better traffic information. Focus on the underlying need. Often a key need is not expressed because it seems obvious. Explore some alternative solutions to get at unstated assumptions.

There is an art to eliciting needs. It involves repeated digging and probing. Ask what they need. Then ask why they need it. Whatever their answer, ask them why they need that. Continue until you are satisfied that you understand what it is they really need and why.

A closer look at a useful tool is *"what if?"* Ask them to consider alternative system approaches. Ask them about alternative technologies, such as cameras rather than loops, or alternative operations, such as local rather than centralized monitoring. This gets at underlying unspoken assumptions, requirements or constraints. Sometimes the stated need is expressed in terms of a familiar solution. For example, the use of the Windows operating system may be cited. Does that mean that an otherwise good Unix-based traffic management system is unacceptable? It may, if the system must interface with legacy Windows systems. These types of questions ferret out the real requirements, and bring previously unstated constraints to light. If you are developing the requirements for your own system, it is helpful to get someone who can think outside the box to probe you in this way and help you clarify your own needs and requirements.

What are sources for a literature search?

This varies greatly from project to project and depending on where in the development process the search is being done. If there is a contract that includes a scope of work, that will be a prime source. If the Concept of Operations has been completed, it will cover needs. Any applicable standards and specifications should be consulted. There may be previous studies for this or neighboring agencies. Other reports, such as strategic plans, will contain information on needs. If multiple agencies are involved, it is essential to understand all such documents.

Suggestions for day-in-the-life studies

If possible, watch them as they perform their jobs. Make note of the sequence of actions (as the basis for scenarios in the Concept of Operations) and anything that appears cumbersome. Then ask them about other situations that you did not see, such as failure events, and how they handle them.

Suggestions for administering surveys

Your local MPO regularly does surveys. Take advantage of their experience. In fact, they are a good source of inputs from your ultimate stakeholder, the traveling public.

Suggestions for one-on-one interviews

By this point you will have put together a description of the needs that you have elicited so far. This is an excellent starting point for discussion. Do they disagree with any of them? Are there any constraints that they know of that would make it difficult to meet the stated needs? Is anything important left out?

Suggestions for workshops

So far you have gathered needs from individuals. Especially when working with multiple agencies, there may be very different priorities and even conflicting needs. For example, a transit agency wants signal priority for its buses, but the agency that operates the roads thinks that it would be too disruptive of traffic flow. The workshop for the first time gets the stakeholders together to listen to each other and to come to an agreement. Maintain an atmosphere that encourages this kind of dialog.

4.8.3 Project Management Practices

OBJECTIVE:

Project management is to integrate all other project activities and tasks so that the goals and objectives of the project are met. Project management practices will plan, execute, monitor, intervene and learn from project activities of each project participant with the goal of completing all project objectives on time, within budget and to stakeholder satisfaction.

DESCRIPTION:

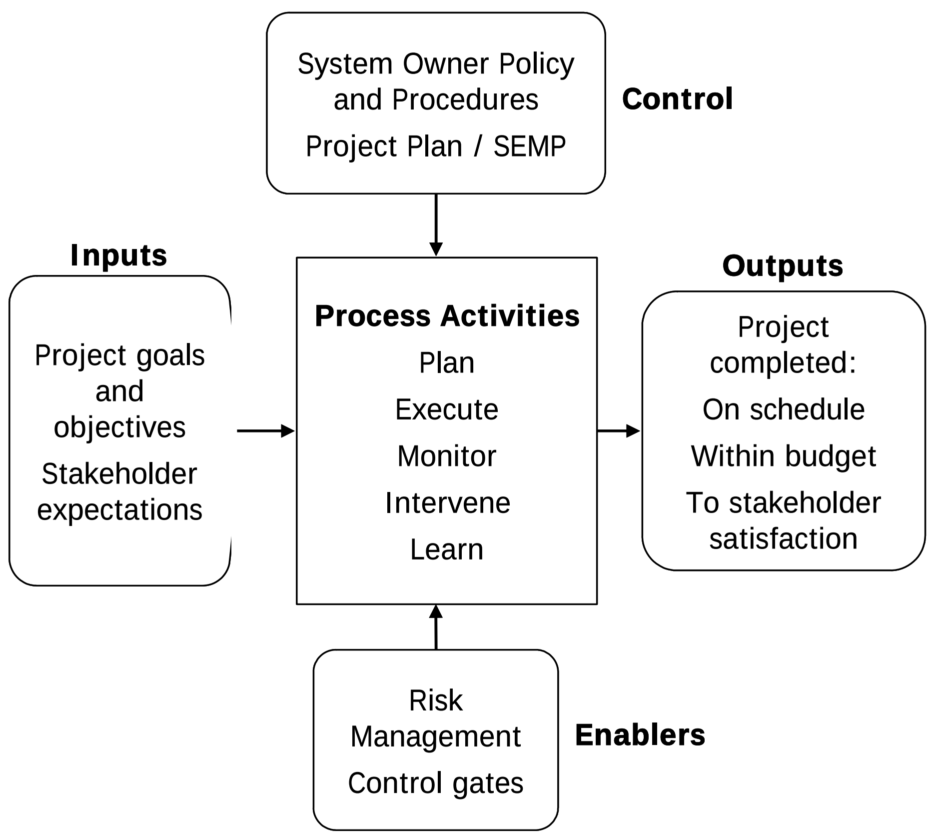
Project management practices include:

- Planning – In coordination with all project participants, developing and documenting the project's plan (task description, budget and schedule) for all necessary project activities
- Execution – Coordinating people and other resources to carry out the project's activities
- Monitoring – Measuring the progress of each project activity according to the plan
- Intervening – Intervening in the execution of an activity to ensure that it continues to support the overall progress of the project
- Learning – Adjusting project management practices based on the experience of previous tasks

Effective communication (written, in meetings and with individual participants) is key to ensuring that project participants are sharing their objectives, status, problems and fixes.

Good project management practices must be married with good management skills for success. These skills include leading (establishing direction, aligning people to that direction and motivating people to overcome obstacles), communicating and stimulating communications among others, negotiating with others on what they need to do, and problem solving with the personnel performing the activity and with their management

CONTEXT OF PROCESS:



PROJECT MANAGEMENT PRACTICES

Inputs:

Project goals and objectives as defined during the initiation of the project by such activities as planning, by the regional ITS architecture and other project studies

Stakeholder expectations as expressed by management, funding providers and internal and external organizations such as engineering, operations, and maintenance.

Control:

System Owner Policy and Procedures will provide both valuable, and sometimes mandatory, guidance from your own agency.

Project Plan /SEMP are prepared during the planning phase of this process and are the basis for management during the remainder of the project.

Enablers:

Risk management is used to stay ahead of the inevitable problems.

Control gates help management measure and ensure progress on the project.

Outputs:

Project completed is the desired outcome of this process, specifically, a project completed on schedule, within budget and to the satisfaction of the stakeholders.

Process Activities:

Plan

Planning the necessary project activities, including a task description, a budget and a schedule, is covered in another section (4.2.1, Project Planning). Planning also looks at needed resources, including e.g., people, stakeholders, and facilities. Planning estimates the amount of work to be done so a budget and schedule can be derived. Planning looks at project risk areas to determine what, if anything should be included in the plan to mitigate those risks.

Execute

Execution is putting the Project Plan / SEMP into motion and ensuring that each activity in the plan is set up to accomplish its assigned tasks. Execution has a lot to do with anticipating the needs of the organization accountable for each activity. Execution is doing everything necessary to make the activities happen as planned and to ensure that the activities do not run into problems which will need after-the-fact intervention.

Monitor

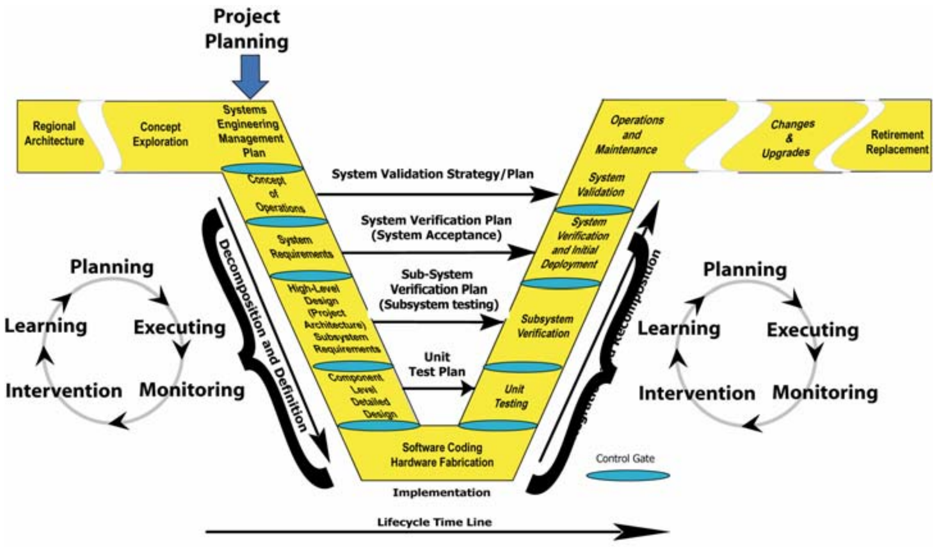
Monitoring involves measuring the progress of each activity to assess its progress according to the plan. In general, activities can be measured by their products, by their expenditures, and by their performance according to the schedule. Expenditures and time are direct measures. More difficult is measuring their progress on products, but if a product can be broken down into parts, then overall progress can be measured by assessing the incremental completion of the parts. Interactive communication with the team is often the best way to get a feel for their progress.

Intervene

When monitoring indicates a problem, project management must act to control and rectify the situation. Intervention may be to the execution of the activity itself or may involve adjustment of other activities to accommodate effects of the problem.

Learn

The Project Plan / SEMP must be considered “living” documents. Progress on the project activities will never go just as planned. The experiences of the preceding activities must be used to modify, as necessary, the remaining activities.



Where does Project Management take place in the project timeline?

Is there a policy or standard that talks about Project Management?

FHWA Final Rule does not specifically mention general project management practices to be followed. CMMI provides best practices in this area.

Which activities are critical for the system owner to do?

In general, project management cannot be delegated to others. Of course, any contractor will be required to have their own project management (which must be defined in their own Project Plan or equivalent). Even then the system owner must still manage the activities of the contractor. Major project management activities include:

- Planning of all project activities along with task description, performing organization, budget and schedule
- Facilitating the execution of each activity, especially by ensuring that all inputs are available and sufficient
- Facilitating the execution of each activity by maintaining open communications between project management and the performing organizations and between the performing organization of related activities
- Monitoring the execution of each activity and intervening in that execution if necessary
- Modifying not only the schedule and budget but the very processes of each activity based on the success of previous activities and encountered risks

How do I fit these activities to my project? (Tailoring)

Planning for project management is one of the most highly tailored of all the project processes. In fact, the purpose of the planning step is to tailor your agency's customary project management practices to the specifics of your project.

It is not uncommon for newer project managers to either over-plan or under-plan their project. With experience, it will become easier to develop project plans that are commensurate with the scope of the project and the usefulness of the information contained in the plan. A few guidelines are:

- Some activities will be routine to the personnel performing the activity and some activities will be new. In general it is best to use existing and familiar processes for the routine activities because the organization will be more comfortable and more efficient doing things the way they always have. For instance, an organization may have their customary processes for managing configuration control of their products. It is generally better to let them use those familiar processes than trying to force them to use new techniques or tools of dubious value. Of course, project management must make sure they will do configuration management when it is necessary.
- The need for detail in the project's plan will increase for activities that involve or impact larger numbers of people, and especially people from different organizations with different management structures. For a small

team of a few people, the need for detail of the processes in the plan can be quite minimal, as long as they understand the products they must produce.

- One area not to skimp on is detail on the deliverables of an activity. These need to be quite clear to the personnel performing the activity.
- The activities covered in the Project Plan / SEMP must align to the technical scope of the project. For ITS, this is especially true for projects needing custom software development. Be sure the plan is developed by people who have experience with the processes needed for each type of product. If the product is software code, then software engineers must be involved in the planning.
- In preparing the project schedule, a careful analysis of each activity's outputs and inputs is necessary to refine the sequence of the activities. Obviously, if an activity needs a certain input, that product must be an output of a previous task. However, it is often possible to initiate an activity before a needed output of another activity is completely finished. In addition to the inevitable start-up tasks, experienced personnel can judge what parts of the previous activity are solid enough to work with.
- A Work Breakdown Structure (see section 4.3.1) is a very useful project management tool to ensure that all tasks have been identified.

What should I track in this process step to reduce project risks and get what is expected? (Metrics)

<u>*On the technical side:*</u>

- Progress on the preparation of activity deliverables and the analysis needed to prepare those deliverables.

<u>*On the project management side:*</u>

- Budget expenditure profiles and the relationship between work accomplished and budget expended
- Task schedules and the similar relationship between work accomplished and time expended

Are all the bases covered? (Checklist)

- ☑ Are the project's goals and objectives clear and do not need to be further defined before project planning can take place?
- ☑ Are the task descriptions, as well as the identification of inputs and outputs prepared for all needed activities?
- ☑ Are the task descriptions, as well as the estimates for cost and time (needed for the budget and the schedule), being prepared by people familiar with the underlying processes?
- ☑ Are the task descriptions, budget and schedule accepted by the performing organizations?
- ☑ Do the financial tracking processes provide you with accurate and timely information on team expenditures?
- ☑ Are you holding regular periodic (usually weekly) meetings with each active task team?
- ☑ Do these meetings review progress on the activity by looking at the preparation of products (outputs), expenditures and progress relative to the schedule?
- ☑ When an activity encounters a problem, are you prepared to intervene in a timely and effective manner?

4.8.4 Risk Management

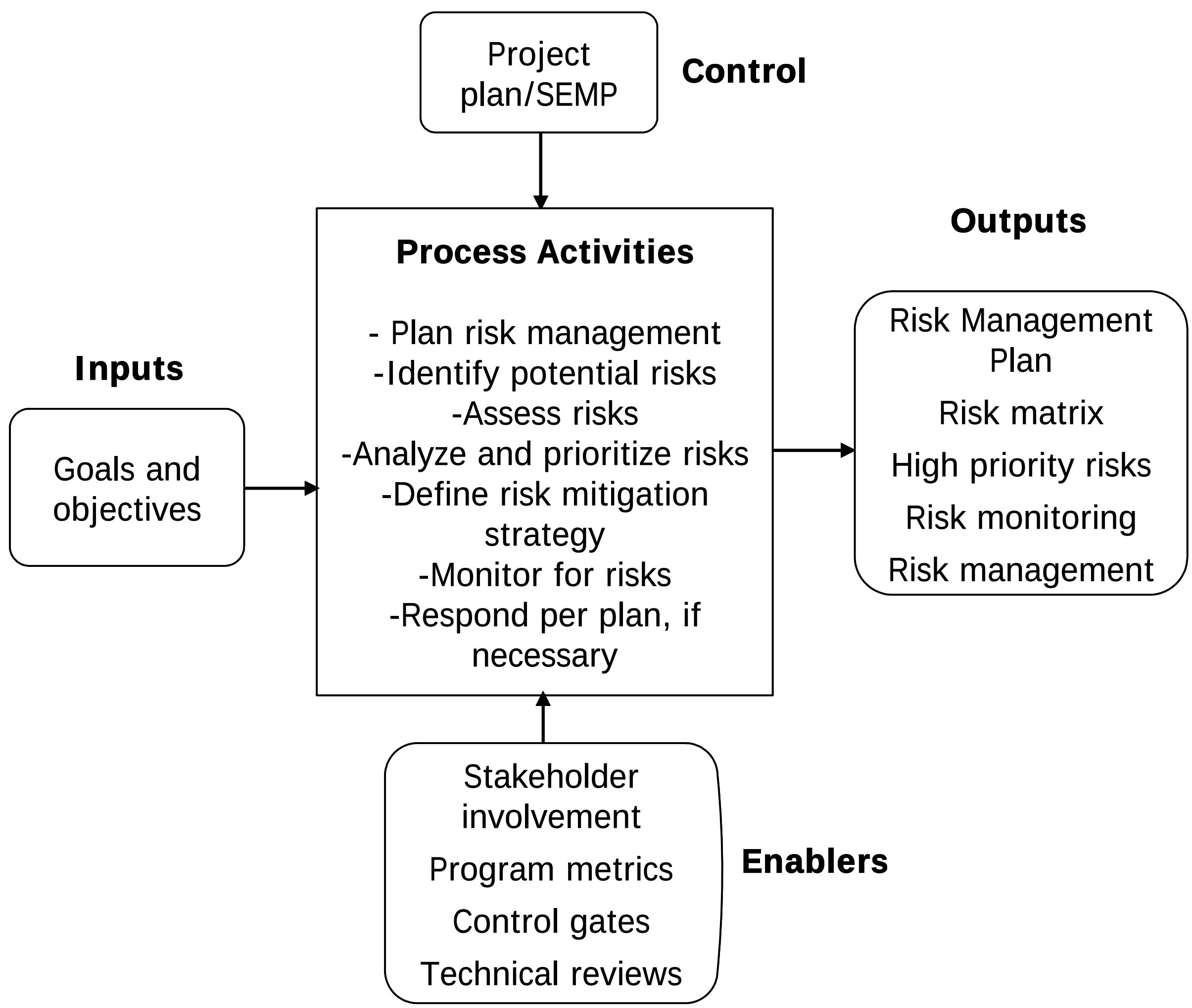
OBJECTIVE:

Risk management achieves a proper balance between risk and reward. It seeks to understand and avoid the potential cost, schedule, and performance risks to a project, and to take a proactive and well-planned role in anticipating them and responding to them if they occur. There are uncertainties involved in any project. The only certainty is that, in at least some small way, things will not go as planned. Risk management anticipates and controls these risks.

DESCRIPTION:

Risk management starts early in the project, by identifying the full range of potential risks. Analysis selects the most critical ones to mitigate or to plan for. It continues throughout the project with monitoring of these potential risks, and a well-planned response to correct problems as they occur.

CONTEXT OF PROCESS:



RISK MANAGEMENT PROCESS

Inputs:

Project Plan defines how the project will be carried out, and so needs to be examined for potential risks
Goals and objectives drive the assignment of risk priorities

Control:

SEMP defines the systems engineering process.

Enablers:

Stakeholder involvement, including participants and outside experts, to identify and rate potential risks.
Program metrics are used for tracking risks
Control gates are structured opportunities to check risk levels and mitigate risk.

Outputs:

Risk Management Plan is the plan on how risk management will be perfomed
Risk Matrix is the graphical representation of the relative probability and consequence of each risk
High priority risks are the top risks to monitor.
Risk monitoring is the ongoing process of tracking symptoms of risks.
Risk management is the ongoing process for correcting any impending problems.

Process Activities:

Plan Risk Management

Develop a risk management plan as part of the Project plan/SEMP. The plan should include risk assessment, mitigation and resolution approach for the project.

Identify potential risks

Stakeholders, project participants, and outside experts first brainstorm potential risks to project success. These should cover all sources of obstacles. It is important to get a broad sample of inputs, since the team faces the greatest risks in areas in which they are not familiar. See the checklist below for potential risk areas. Collect "lessons learned" from previous projects to help identify potential risks.

Assess risks

There are two components to risk—the likelihood that an undesirable event will occur and the consequences if it does occur. Likelihood is expressed quantitatively as a probability percentage, or qualitatively in terms of categories such as likely, probable, improbable, and impossible. Consequences may be expressed quantitatively in terms of dollars or performance metrics, or qualitatively in terms of categories such as catastrophic, critical, marginal, and negligible.

Analyze and prioritize risks

The risk of an undesirable event is based on these components. Quantitatively, risk is the expected value of a potential loss, based on reasoning under uncertainty. Qualitatively, risk is represented by positions in a risk matrix. The risk matrix is useful in both types of analysis. The columns represent the likelihood, and the rows the consequences. Anything that falls in or near the "likely/catastrophic" box is high risk. Start in that corner and select the top risks of concern. (See Tip below)

Define approaches to handling the top risks

Identify and evaluate alternatives for handling the top risks. Even before the monitoring indicates a problem, there are steps that can be taken to control the high-priority risks, such as changing things to eliminate them, reduce their likelihood, or reduce their impact. One example is eliminating requirements that carry a high risk but are of marginal value. Another is parallel development. Plan contingencies for the remaining highest priority risks before starting, and then monitor them regularly.

Monitor risks

Identify metrics for each of the selected top risks. As a management tool, these are the triggers that release contingency funds to address a problem. Be sure that these metrics are easy to track and signal a potential or imminent problem. Cost and schedule are always risks, and their metrics are spending and performance to schedule (see 4.8.5). Set up a schedule and procedure to track the metrics on a regular basis.

Respond per plan, if necessary

If the monitoring indicates a problem you will be able to respond quickly, since there is a plan in place. This avoids the common problem of poor decisions and project redirections under the pressure of the moment.

Where does risk management take place in the project timeline?

Is there a policy or standard that talks about Risk Management?

FHWA Final Rule does not specifically mention general risk management practices to be followed. CMMI provides some best practices in this area.

Which activities are critical for the system owner to do?

- Participate in risk identification
- Review and approve the identified key risks
- Ensure ongoing risk monitoring
- Participate in mitigation activities
- Lead the development of the risk plan

How do I fit these activities to my project? (Tailoring)

The level of each activity should be appropriately scaled to the size of the project. For example, a small project may consider only a few risks and prioritize them qualitatively. The level of intensity of monitoring and mitigation should be appropriate for the project risk. A project that is technically and organizationally similar to previous ones may need only to monitor cost and schedule.

What should I track in this process step to reduce project risks and get what is expected? (Metrics)

On the technical side:

- Number of potential risks in each of the higher risk categories (e.g., frequent/catastrophic)
- Risk monitoring metrics as established in each step

On the project management side:

- Completion of documentation of risks and their priorities
- Number of high priority risks with a documented resolution plan

Are all the bases covered? (Checklist)

- ☑ Is the risk management plan included in the Project Plan/SEMP
- ☑ Have all sources of risks been identified?
  - ☑ Technical (e.g., new detectors do not perform as expected)
  - ☑ Institutional (e.g., agency data sharing, new regulations, public opposition)
  - ☑ Funding (delays or cuts)
  - ☑ Environmental (e.g., weather for field equipment, restrictions on building)
  - ☑ Personnel (e.g., loss of key personnel, substandard performance)
  - ☑ Commercial (e.g., vendor does not deliver COTS product)
- ☑ Were experts and stakeholders queried in all the areas of risk to develop a broad list of credible risks?
- ☑ Are the risks prioritized and the most critical ones identified?
- ☑ For each high priority risk, are there ways to eliminate the risk, or reduce its likelihood and/or impact?
- ☑ For each high priority risk, have the symptoms of the problem and a means for monitoring them been identified?
- ☑ Are the high priority risks regularly monitored throughout the project?

☑ For each high priority risk, is there a risk resolution plan?

Are there any other recommendations that can help?

All useful systems incur some risk. The goal is a balance between system performance and risk. That is why the focus is on only the most critical risks. Lesser risks will and should be accepted.

From a management viewpoint, there are four ways to handle risk. 1) Mitigate the risk by allocating contingency funds to its resolution if it becomes necessary, 2) accept a risk that cannot realistically be mitigated (such as an earthquake), 3) avoid the risk by change of requirements or design, and 4) transfer the risk (for example to an insurance company or to a developer under a fixed price contract).

Even if you have a dedicated risk management team, ***everyone on the team must be encouraged to identify potential risks***. A "shoot the messenger" atmosphere will only allow hidden risks to grow out of control.

Uncertainty is what makes risk management both difficult and essential. There are statistical techniques, such as probabilistic decision theory for reasoning under uncertainty. The most basic technique is expected value. Risk is computed as the probability of failure multiplied by the consequence of failure. Probability is between 0 (impossible) and 1 (certain). Consequence is ideally expressed in terms of dollars, specifically the recovery cost. Since this is difficult to estimate, you may rate each consequence between 0 (negligible) and 1 (catastrophic). The product of probability and consequence then gives a risk metric between 0 (no risk) and 1 (definite and catastrophic). For more information, see INCOSE Systems Engineering Handbook, July 2000, section 4.2.4

TIP ***Don't worry if you don't have the data to compute expected costs and probabilities for risks in prioritizing them.*** You can get reasonably good results simply by rating risks qualitatively relative to three to five categories in each of impact and likelihood. Below is an example of the matrix used for such an evaluation. The numbers are the order in which the risks are to be considered. Anything that is in the box labeled "1" is your highest priority. In fact, any risk that is both catastrophic and likely indicates a serious system problem requiring a change in requirements or design.

	Likely 0.7 - 1.0	Probable 0.4 to 0.7	Improbable Less than 0.4, but not 0	Impossible 0
Catastrophic 0.9 - 1	1	3	6	
Critical 0.7 - 0.8	2	4	8	
Marginal 0.4 - 0.6	5	7	10	
Negligible 0 - 0.3	9	11	12	

A closer look at definitions and examples of consequence and probability ratings

Here are definitions to firm up the consequence levels used in the matrix (from INCOSE Systems Engineering Handbook). Here the "mission" is the purpose of the system, such as traffic management.

Catastrophic: Failure would result in project failure meaning a significant degradation/non-achievement of technical performance.

Critical: Failure would degrade system performance to a point where project success is questionable for example, a reduction in technical performance.

Marginal: Failure would result in degradation of a secondary system functions, a minimal to small reduction in technical performance.

Negligible: Failure would create inconvenience or non-operational impact. No reduction in technical performance.

Here are examples of some of the characteristics that would impact the probability of failure (adapted from INCOSE Systems Engineering Handbook).

Maturity:

- Existing system, probability is 0.1
- Minor redesign, 0.3
- Major change (feasible), 0.5
- Complex design (technology available), 0.7
- State of the art (some research done) 0.9

Complexity:

- Simple design, 0.1
- Minor increase in complexity, 0.3
- Moderate increase in complexity, 0.5
- Significant increase in complexity, 0.7
- Extremely complex, 0.9

Note that if there are multiple risks, the overall probability will be at least as high as the highest of them, and often even higher.

4.8.5 Metrics

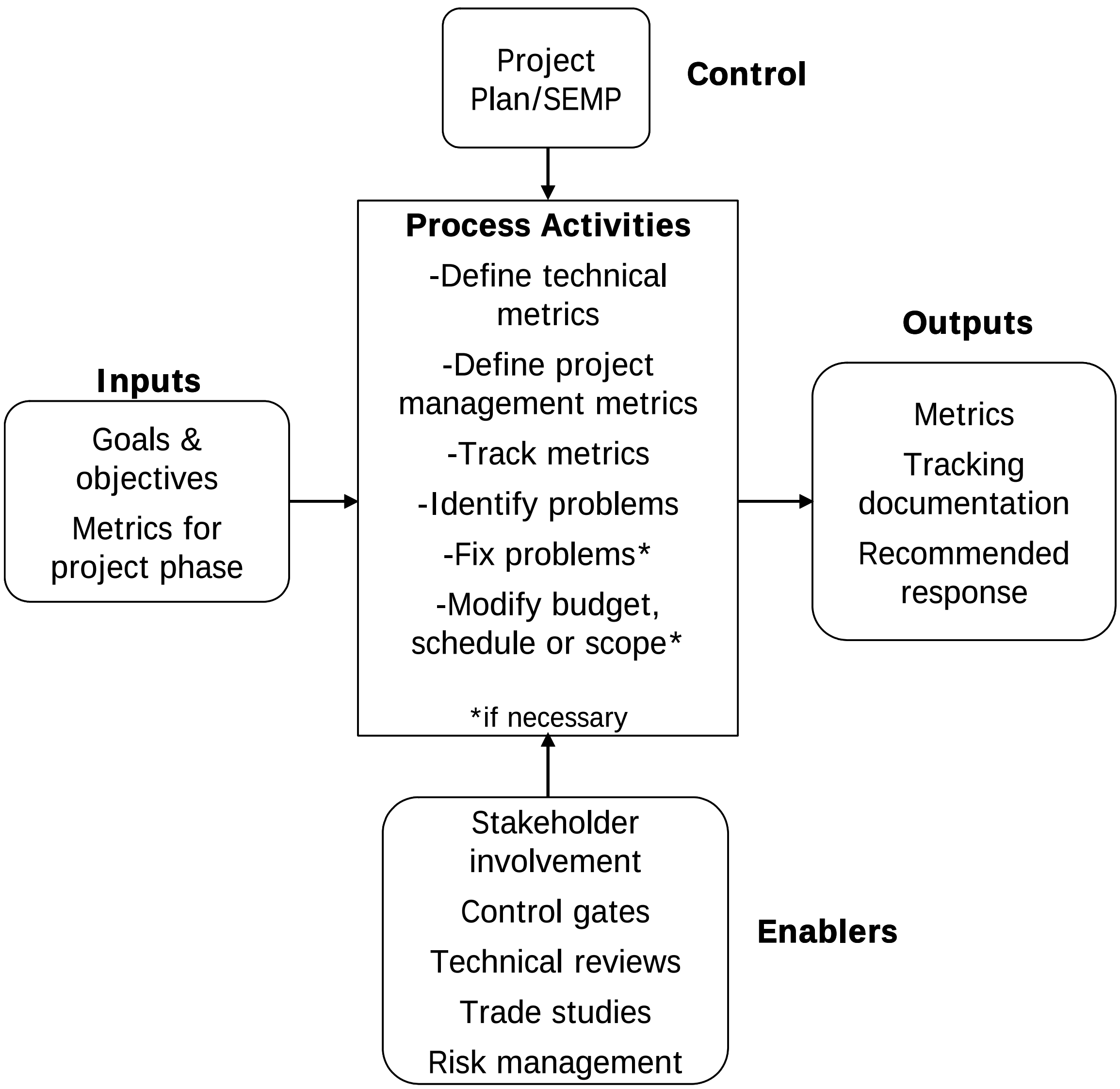
OBJECTIVE:

Metrics are a tool to recognize and correct problems as soon as they occur. Good metrics are meaningful (i.e., they represent the progress of the project or expected performance of the system), easy to collect, and help make a decision.

DESCRIPTION:

There are both technical and project management metrics. Technical metrics track how well the finished project will meet its performance objectives. Project management metrics are whatever should be tracked during each process step to reduce project risks and get what is expected. Cost and schedule are key project management metrics for any project. Other metrics may come out of the risk management process or performance requirements. Each activity described in Section 4 includes a list of suggested metrics to be tracked during or in support of that activity.

CONTEXT OF PROCESS:



METRICS PROCESS

Inputs:

Goals and objectives define direction, priorities, and change of course, if necessary.

Metrics for project phase are defined by each project step to check its progress.

Control:

SEMP provides guidance for the systems engineering process.

Schedule and *Budget* constrain the project.

Enablers:

Stakeholder involvement enables the stakeholders to suggest metrics and provide guidance.

Control gates provide an opportunity for tracking progress.

Risk management suggests and uses metrics.

Outputs:

Metrics are the selected measures of project progress and performance

Tracking documentation is a history of project progress relative to the metrics.

Recommended response to noted project problems documents the recourse and rationale.

Technical reviews suggest metrics and review their tracking.

Trade studies compare alternatives relative to the metrics

Process Activities:

Define technical metrics

Technical metrics track expected performance and effectiveness of the system being developed. These are related to the system mission. They often address critical performance parameters such as response time or accuracy. An example is the time to compose an Amber Alert message and get it displayed on CMS system-wide, since this must be done within 15 minutes. Each of the sections in this section also includes technical metrics related to the task, such as an increase in high-risk requirements.

Define project management metrics

Project management metrics track progress. Define metrics that indicate a potential problem. These then act as triggers for risk management. Each step of the systems engineering process has metrics associated with it. Each task in Section 4 includes a list of project-related metrics specific to that task. These give an indication of how far along the task is. For example, concept selection has metrics for the percentage of candidate concepts evaluated and the percentage of stakeholders who have approved the study. In some cases the metrics will simply be whether or not something has been completed. There are other metrics related to milestones or to how much has been developed or delivered, such as number of subroutines written or lane-miles instrumented. Appropriate milestones should be meaningful and easy to track. Spending to date is always an essential metric.

Track metrics

Progress on the milestones and status of metrics relative to cost and schedule is compared regularly with the expected progress. A simple way to do this is to plot the metric relative to dollars spent. The best estimate of where the project will end up comes from extrapolating the line out to the end. This is because, despite all efforts to “catch up,” projects that are behind tend to continue to fall behind by the same percentage. If progress is found to be lagging, a response will be carried out to correct the problem and get the project back on track.

Identify problems

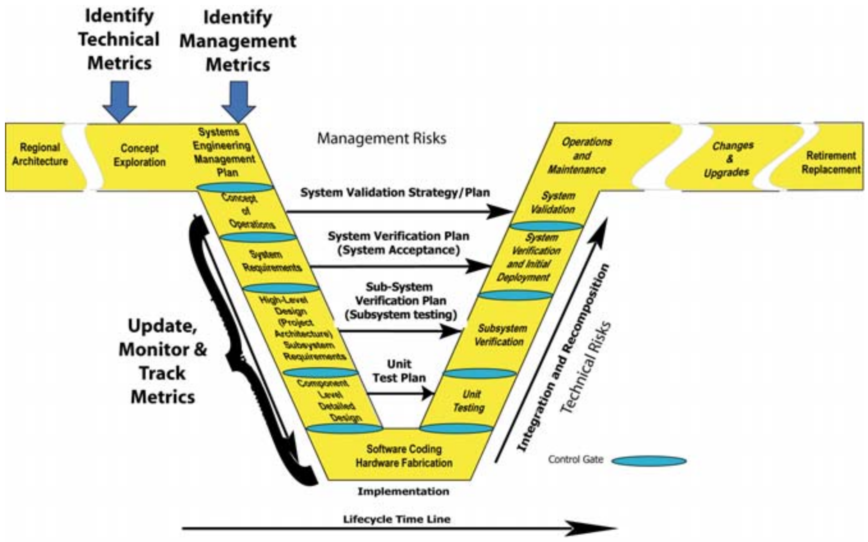
If any of the metrics indicate a potential problem, use them to trace back to the source of concern. A fast and decisive response is necessary to get a project back on track. Asking the right questions will suggest what action to take. Are there any problems that are hindering development? Do we have the right resources? Is the budget realistic? Is the schedule realistic? Is the scope overly ambitious?

Fix problems, if necessary

Make sure personnel are properly assigned and well used. Eliminate any efforts that do not trace back to the requirements, and so are unnecessary.

Modify budget, schedule, or scope, if absolutely necessary

Sometimes project plans are overly optimistic. If you’ve done everything possible to streamline the project and solve problems, it may be that the scope of work cannot be completed in the time and budget allocated. Sometimes hard decisions need to be made, and will only get worse if delayed. Consider eliminating requirements of marginal value, extending the schedule, or asking for more budget. This is extremely disruptive, and should be avoided if at all possible. Ideally, the project plan is carefully and realistically developed up front, so that these types of changes are avoided.



Where do metrics take place in the project timeline?

Is there a policy or standard that talks about Metrics?

FHWA Final Rule does not specifically mention practices to be followed for metrics. CMMI has material related to metrics

Which activities are critical for the system owner to do?

- If known, Identify critical metrics to track
- Review metrics tracking
- Plan and carry out a response if the metrics indicate one is needed

How do I fit these activities to my project? (Tailoring)

The level of each activity should be appropriately scaled to the size of the project. For example, a small project will have fewer and less complex metrics. A larger project will more likely use earned value (see Closer Look, below) or similar techniques. Always track spending.

What should I track in this process step to reduce project risks and get what is expected? (Metrics)

On the technical side:

General technical metrics are based on the mission of the system being developed, and will be defined as part of this task. Examples are

- Incident response time (for an incident management system)
- Time for a composed message to appear on a changeable message sign (CMS system)
- Accuracy of speed estimates (traffic monitoring system)

In addition, each project task has its own technical metrics, as described in this section.

On the project management side:

- Expenditures to date
- Progress to date (actual metric is project-specific)

In addition, each project activity will have its own metrics. The specific metrics used will be selected as part of this activity.

Furthermore, this activity has metrics of its own:

- Percentage of planned metric tracking that is actually carried out.
- Percentage of identified problems that are subsequently resolved.

Are all the bases covered? (Checklist)

☑ Are the metrics good indications of the progress of the project?

☑ Are the metrics meaningful and clear?

☑ Are the metrics easy to collect?

☑ Do the metrics support making a good decision?

☑ Is the number and complexity of the metrics reasonable for the size and complexity of the project?

☑ Are the metrics tracked regularly?

☑ Is there a plan for identifying and responding to a lagging project?

Are there any other recommendations that can help?

If your project is behind schedule early on, the temptation is to think that you will catch up by "working harder." In fact, studies show that projects tend to lag by the same percentage throughout their lifetime. For example, suppose you have a $100,000 project and you've spent $20,000 but only progressed to where you would expect to be after spending $15,000. You are only $5,000 behind and may feel that you will catch up. In fact, what usually happens if you don't change anything is that you are 25% behind, and will tend to remain 25% behind, and only be 3/4 done when the money is gone.

A closer look at Earned value analysis is a technique for comparing actual and expected progress to date, and for estimating future progress. The earned value is simply the project budget multiplied by the percentage completed. For example, if you are putting in a traffic signal system, a good metric could be the number of intersections completed. If you have, for example, completed 10 out of 50 intersections, you are 10/50 = 20% done, and would have expected to have spent 20% of your budget. That is your earned value. If you then divide this by what you have actually spent to date, you get the performance ratio. If it's greater than one you are in good shape, but if it is less than one you are overspending.

You can also estimate what it will cost to finish the project (cost to completion). Divide the spending to date by the percentage complete. Compare this with your budget.

In the example above, the project has a $100,000 budget and has spent $20,000 to do $15,000 of work (15% of the job). The earned value is $15,000, and the performance ratio is 15,000/20,000 = .75. This is less than one, and in fact it indicates that the project is 25% (1 - .75) behind. The estimated cost to complete is $20,000/.15 = $133,333, well above budget. This project needs to make changes.

It is generally difficult to estimate percentage completed. We have all known developers who were "90% done" during most of the project duration. Hence a simple and more objective approach is to earn value only where there are clear milestones, specifically at the beginning and end of each task. Rather than try to track progress on a small task, call it 100% only when it is complete, but 0% before that. A longer task would have a 50% earned value for getting started, and 100% only at the end. Combining these over many tasks gives a good indication of overall project progress.

Combining metrics Generally the alternatives are compared relative to multiple metrics that have very different sizes and units, such as number of vehicles per hour, average speed, number of incidents, time to respond, mean time between failures, cost, rating on a scale of 1 to 5, and so on. These must be normalized to a common scale before they can be combined in any meaningful way. There are many ways this can be done, but the following ratio seems to give the most reasonable results. It compares expected performance of the candidate system to current performance.

(ideal – current)/(ideal – candidate).

For example, if the current system is 96% reliable, and the candidate is 98% reliable, here is how it will come out. The ideal is 100% reliable, so we get (1.00 -.96)/(1.00 - .98) = .04/.02 = 2. This agrees with our intuition that a 2% failure rate is twice as good as a 4% failure rate. This ratio works even if the metric is negative, i.e. less is better. For example, if there are currently 300 major accidents a year and simulation shows that a candidate system will average only 200 accidents a year, then since 0 accidents is the ideal, we get (0 – 300)/(0 – 200) = 1.5.

Once all of the metrics are normalized in this way they can be combined. Assign a weight to each of the metrics. Be sure the weights add up to 1. The weight is an indication of the relative importance of each of the metrics. For example, for a high-level freeway concept with metrics based on safety, capacity, and public acceptance, the weights could be respectively .5, .25, and .25, indicating that safety is most important. Multiply each of the normalized metrics by its respective weight and add them together. The result is a unit-less measure of the goodness of each of the alternatives relative to the current system.

Lifecycle cost can be included in this calculation, but it is often more meaningful to separate it and plot overall effectiveness against cost. This allows you to take into account budget constraints, and to identify good low cost and high cost solutions, possibly with an evolutionary path between them.

4.8.6 Configuration Management/Interface Management

OBJECTIVE:

Configuration management ensures that project documentation accurately describes and controls the functional and physical characteristics of the end product being developed (establishing product integrity). The second part is to synchronize changes to the system with its documentation. This occurs throughout the system lifecycle (maintaining product integrity).

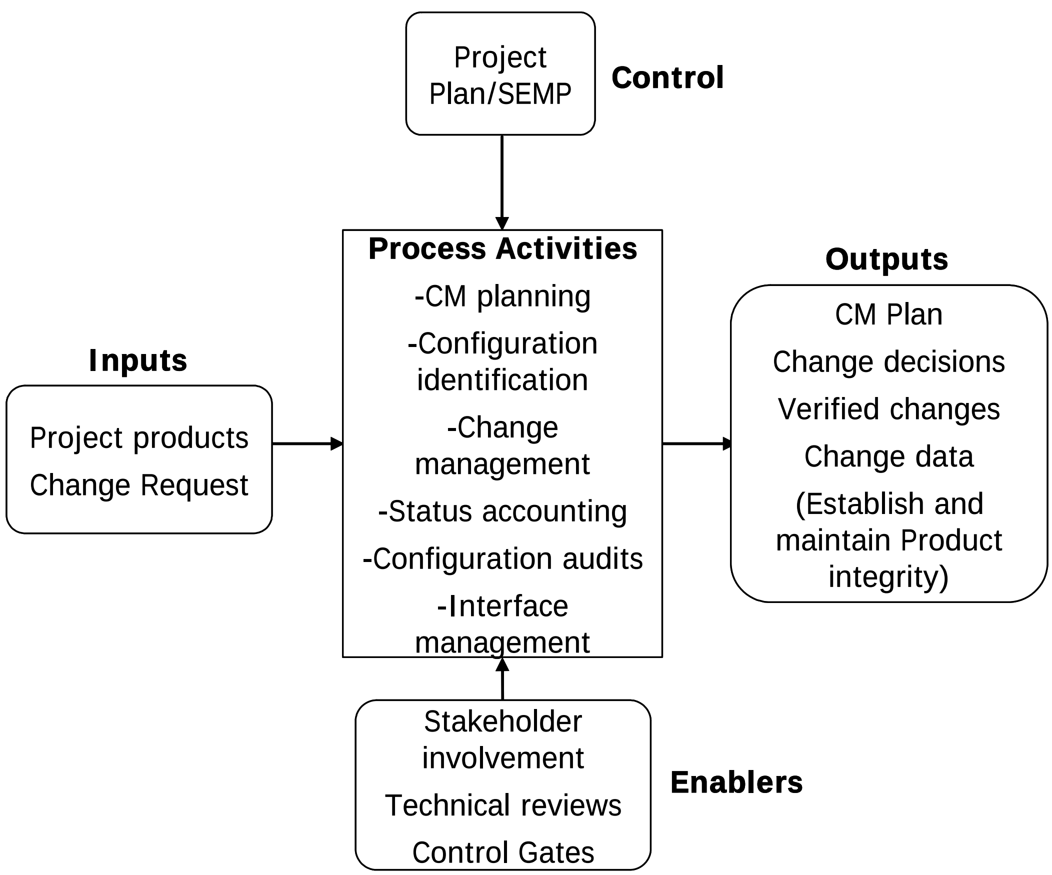
DESCRIPTION:

Configuration management (CM), in conjunction with other systems engineering activities, is used to establish system integrity (integrity is defined as all system functionality, physical characteristics and design match its documentation) and then maintain this integrity throughout its life. As illustrated, configuration management is an integrated set of activities that include:

- CM planning and management – Plan the configuration process and manage it.
- Configuration identification – Identify the items to be placed under change management
- Change management – Manage changes to the configuration items
- Status accounting – Track change information
- Configuration audits. – Verification that the changes match the documentation

Interface management is a key configuration management practice that has a focus on interfaces. Since interfaces give the system leverage and access to stakeholders there should be special attention paid to them. Usually a specialized group (or individual) called the interface control working group is established to just manage interfaces.

CONTEXT OF PROCESS:



CONFIGURATION MANAGEMENT

Inputs:

Project Products that have been approved to be managed under the CM process throughout the life of the system.

Change Request(s) for products under change management control.

Control:

SEMP/Project Plan will contain the CM plan(s) for the system owner, and development team.

Enablers:

Stakeholder involvement in the change management board and in changes that affect them

Technical reviews are used to evaluate changes prior to submission to the CM board

Control Gates are used to establish baseline products that allow the project to move forward to the next phase of work.

Outputs:

CM Plan contains the process needed to carry out CM for the project

Change Decisions on change requests

Verified changes after implementation and the synchronization of the documentation

Supporting change data that identifies the change and rationale for the change

Process Activities:

Plan configuration management activities

There are three application areas that need planning. The agency's CM plan for the life of the system, the implementation team's CM plan for development and the vendor supplying commercial off-the-shelf products. The agency's CM plan should identify the requirements for the development team's CM plan and vendor's CM plan and the needed outputs to support the life of the system.

Identify configuration items

This step identifies items that will be managed under the CM process. These are called "Configuration Items (CI)". These items exist at all levels of decomposition and occur in each phase of development. For example, baseline requirements and design documents developed during the definition and decomposition of the system are configuration items. When products are identified, e.g. subsystem at the detailed design level, and when the end products of software and hardware are complete functional units, these are product level configuration items. Finally, when the system is deployed, the operational baseline is a configuration item.

Manage change

This is the process to manage changes to the configuration items. This involves a change management board and documentation that identifies the change, rationale, cost, risk and priority.

Perform status accounting

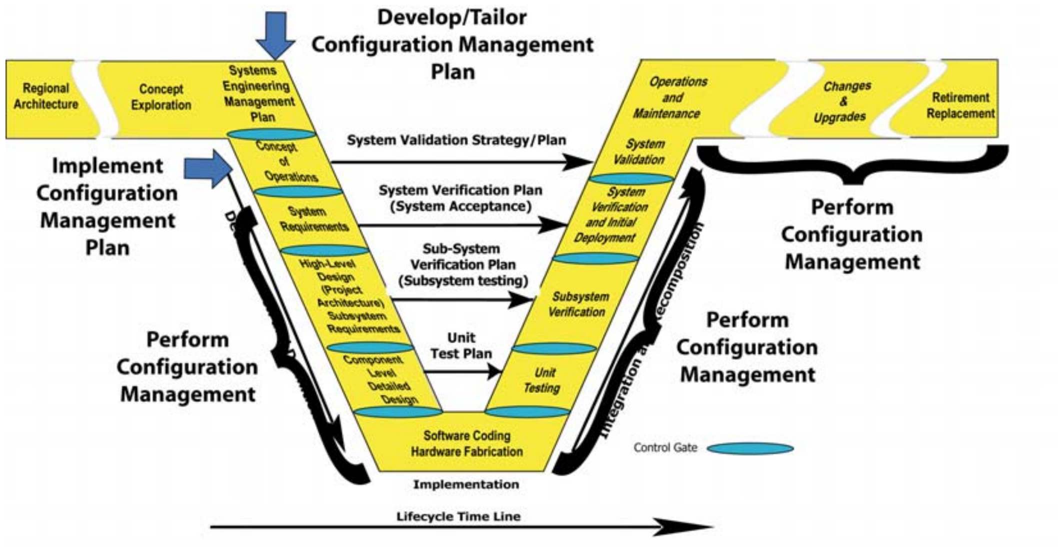
This step collects change data and is used for status and analysis purposes.

Perform configuration audits

There are two types of audits (functional and physical audits). Functional audits match the product to the functional and performance requirements (acceptance verification) and physical audits match version numbers and physical identifiers with the documentation.

Manage interfaces

This step manages the interfaces of the system. This activity controls both external and internal interfaces. Interfaces that are shared with other agencies require an Interface Control Document that is an agreement as to the specification of that interface.



Where does the Configuration Management take place in the project timeline?

Is there a policy or standard that talks about Configuration Management?

FHWA Final Rule does not specifically mention general Configuration Management practices to be followed. EIA 649 National Consensus Standard for Configuration Management and the Mil-Hbk-61 provide a great deal of application information.

Which activities are critical for the system owner to do?

- Establish a CM process for the project.
- Participate and chair the change management board.
- Gain stakeholder support for the project configuration management process.
- Initiate periodic CM audits to maintain confidence in the integrity of the system.
- Review the vendor's and development team's CM processes

How do I fit these activities to my project? (Tailoring)

The CM process is scalable to the level of custom development that is being done. For systems that are all COTS products, the primary CM activity is a review of the vendors' CM processes to ensure that the vendor will provide appropriate updates to the product and notices when the product changes or is discontinued. This continued support most likely will require an on-call service contract and a warranty period. On the other end, where the development is a large multi-regional system with multiple stakeholders, and a mix of custom hardware/software and commercial products that is owned by the system owner and stakeholders, the CM process will be more involved.

What should I track in this process step to reduce project risks and get what is expected? (Metrics)

On the technical side:

- Changes to specific area of the system. A high number of changes may indicate a design weakness.
- Monitor the impact of a change: who will be affected / how much of the system will need to be changed? A seemingly small change may have an extensive impact and it will be the impact of the change that will determine the cost and risk of change.

On the project management side:

- Growth in the number of change requests. This is an indication that the baseline was established too early.
- Monitor the types of changes. Determine if the changes are critical to meet the initially stated requirements or if this is new functionality that can be deferred to the next phase of work.

Are all the bases covered? (Checklist)

- ☑ Is there a CM plan for the project?
- ☑ Was the plan reviewed and supported by all the stakeholders?
- ☑ Is the organization for CM in place for the project?
- ☑ Is there sufficient funding to sustain the CM activities throughout the life of the system?

- ☑ Does the development team have a CM process and was it reviewed by the system owner and stakeholder?
- ☑ ***<u>Is the product documentation complete</u>*** to the extent that the system owner can use another qualified development team to upgrade and maintain the system independent of the initial development team? **(extremely important)**
- ☑ Does the COTS vendor have a CM process for their products?
- ☑ Does the vendor provide a notice of design changes?
- ☑ Does the vendor provide a notice of obsolescence?
- ☑ Does the vendor provide on-going maintenance support?

Are there any other recommendations that can help?

Configuration management for systems development is a management process for the project products. Configuration management works together with a good systems engineering process. The systems engineering process provides the orderly establishment of the project products and documentation. Configuration management provides for change management and synchronization of the changes to the product with its documentation.

Use configuration management as an evaluation tool and discriminator for vendors of commercial-off-the-shelf products (COTS and development teams). COTS products for Intelligent Transportation Systems should have a long life through upgrades. A vendor that has a good internal CM process can show how their products are maintained and upgraded. The vendor would not only maintain and upgrade their products on a regular basis, but issue notices of design changes, and notices of obsolescence when products reach their end of life. This type of service would most likely be part of an on-call service contract.

Configuration management should not be assumed as part of the project. It must be planned. The cost of Configuration Management on large projects is estimated at around 5% to 10% of the lifecycle costs. This data was from an informal poll of systems engineers from the aerospace and defense contractors. This cost covers starting a CM activity from the ground up (10%) or using an existing in place CM process and extending it to include a new project (5%). Once a CM process is in place it should be used for future projects as well.

Internal interfaces may need Interface Control Documents (ICD) if the interface is shared with another system not under control of the system owner.

A closer look at three different environments for configuration management

System owner has the ultimate responsibility for the system over its life. Various development teams and vendors may be involved in providing or developing products for the system and in providing upgrades, maintenance and evolution of the system over its life. The system owner should have a CM process that covers the life the system. The vendors and development teams working on the system should provide the products and documentation that will be compatible with the system owner's CM plan.

Development Team(s) should have their own configuration management processes and tools when developing hardware and software for the system. This CM process addresses the low level procedures needed when software and hardware is developed. The system owner should define the expected output from the development team's process but should not dictate how the development team performs CM. However, inspection of the team's CM process should show that it conforms to industry standards (see references for a list).

COTS Vendors should have internal CM processes that are documented and followed and should share these documented processes with the purchaser of their equipment. At a minimum, the vendor should maintain a configuration profile of the version of the product sold. This profile identifies the version of software and hardware used and the matching documentation. The vendor should define the expected support life of the product, provide notices of design changes and obsolescence and provide product updates with revised documentation. These additional services may be available free over a warranty period. For extended periods of support they most likely will be at an additional cost.

4.8.7 Process Improvement

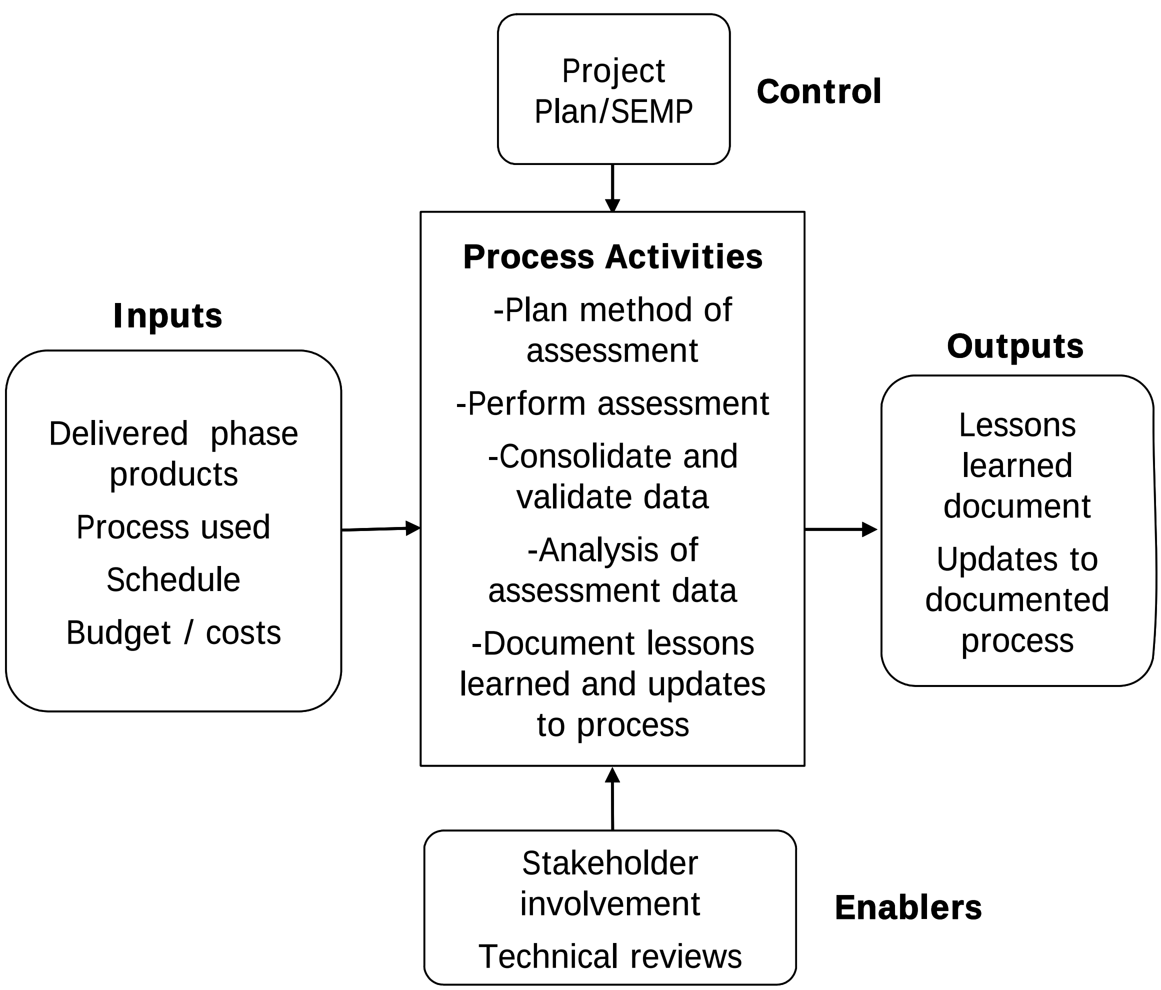
OBJECTIVE:

Process improvement provides a method for the continuous improvement of processes and products over the lifecycle. These process improvements are transferable to future projects (development, operations and maintenance, and retirement/replacement).

DESCRIPTION:

At the completion of each phase of a project lifecycle, the processes and the quality of products delivered should be reviewed, assessed and documented. At the completion of the project, the assessment for each phase should be reviewed and summarized as to its impact on the success or shortfalls that occurred during the project. The scope of the assessments should cover processes (methods and techniques) used during the performance of each phase of the project, the quality of products produced, and the stakeholders (system owner, consultants, vendors, and development teams) that were involved. Once documented, the assessment is used to improve the processes in place. Documented lessons learned will capture the "corporate" knowledge gained from the experience of the project so that the lessons learned can be applied to remaining phases of the existing project and future projects. The assessment has three primary activities, the planning, strategy and performance of the assessment. This assessment should be an on-going part of each project. This can be performed by the system owner, other stakeholders, or an independent assessment team.

CONTEXT OF PROCESS:



PROCESS IMPROVEMENT PROCESS

Inputs:

Delivered phase products delivered products for the phase of work.

Process used actual process that was used for the phase of work.

Schedules that were used for the phase of work including the original schedule and any updates

Budget/costs that was developed for the phase of work and any updates.

Control:

Project Plan/SEMP contains the process used to carry out each phase of the work and to determine the process improvements for the project.

Enablers:

Stakeholder involvement is essential in process improvement. The stakeholders include the system owner, development team, consultants, and all direct stakeholders that were involved in the phase of work.

Technical Reviews process is used to carry out the workshops and interviews stakeholders on possible process improvements.

Outputs:

Lessons Learned Document for each phase of work will identify the strengths and weaknesses of the process and the products of the phase of work.

Updates to documented process will be developed as a result of the assessment, and recommended changes to existing documented processes.

Process Activities:

Plan Method of Assessment

Identify the set of assessment activities needed for the phase under assessment. Identify roles and responsibilities and the timeline for the assessment.

Identify the purpose for the assessment, scope of assessment and assessment team qualifications, what team members should be interviewed, and what phase products and processes need to be assessed.

Perform Assessment

The methods of assessment include interviews, questionnaires, surveys, and workshops. What criteria are established for the documents need to be reviewed. Example assessment would compare the planned processes to the actual processes, and the planned deliverables with the actual deliverables

Consolidate and validate data

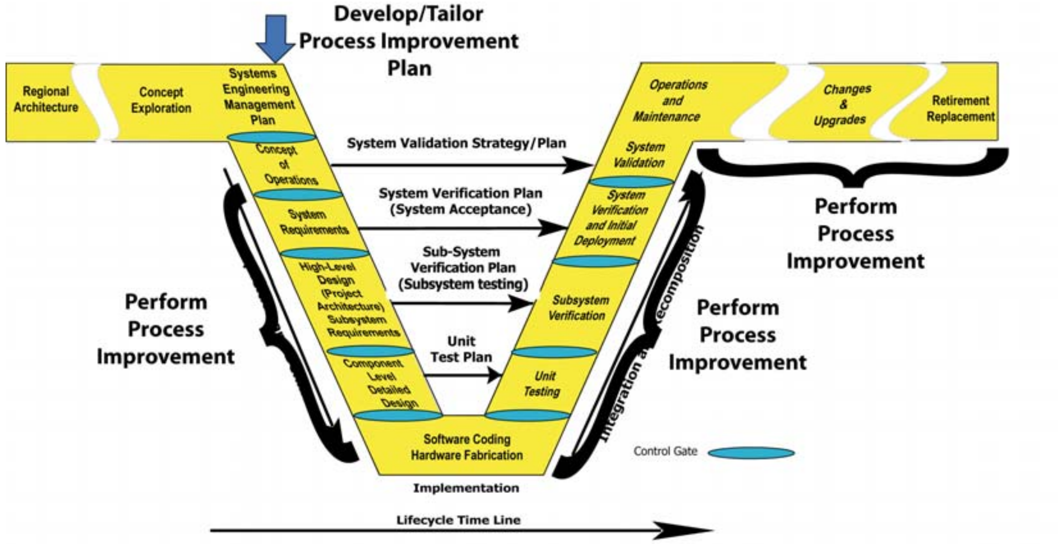
Identify the constraints on the way in which an assessment method provides for the consolidation and validation of the data collected. Collect and consolidate the data from observation of products and processes in a way that can lead to unbiased conclusions and accurate observations. One test of this is that it would be a repeatable observation and validation of results.

Analyze assessment data

Reach assessment team consensus on findings, ratings, and validated observations from a minimum of two different sources. Identify strengths and weaknesses of the product and process under assessment.

Document lessons learned and updates to process

Document the lessons observed and reported from the assessment. Develop a plan to update the process and migrate to practice. Clearly linking the lessons documented to the upgrade of documented processes and clearly observed change in practice is Lessons Learned.



Where does the Process Improvement take place in the project timeline?

Is there a policy or standard that talks about Process Improvement?

FHWA Final Rule does not specifically mention general process improvement practices to be followed. CMMI contains the information for process improvement and assessment.

Which activities are critical for the system owner to do?

- Lead development of the framework for process improvement for the project and include it as a SEMP item.
- Participate in the performance of process improvement interviews and workshops.
- Gain stakeholder support for the process improvement activities.
- Lead updating of the process improvement documentation as appropriate for the system owner.

How do I fit these activities to my project? (Tailoring)

Process improvement for small projects such as a traffic signal system upgrade using a single vendor can be done one time at the completion of the project. For large projects that may involve consultants and development teams for different phases of the work an assessment is recommended after each phase of work. The purpose is to capture any lessons learned as early as possible while the project activities are fresh in memories of the stakeholders involved and before the development team leaves the project. Once documented for each phase, an overall assessment is recommended when the project is completed.

What should I track in this process step to reduce project risks and get what is expected? (Metrics)

On the technical side:

- Assess that the correct technical metrics were defined for the project.
- Assess that the correct technical process was used to carry out the work for the phase.
- Assess the difference between the expected results and the actual results.

On the project management side:

- Assess the cost for collecting the technical metrics.
- Assess the cost for using the process to carry out the work.
- Assess the difference between the planned baseline cost/schedule and the actual cost/schedule.

Are all the bases covered? (Checklist)

☑ Was a process for the assessment developed as part of the project plan/SEMP?

☑ Were the key stakeholders identified for the assessment?

☑ Were the key stakeholders for the phase interviewed as to the strengths and weaknesses in the performance of the phase of work?

☑ Were meeting minutes and notes kept for the assessment?

☑ If a key stakeholder leaves the project, were they interviewed on the strengths and weaknesses of the phase of work?

- ☑ Is there a plan for on-going process improvement throughout the operations and maintenance phase of the project?
- ☑ Is there a defined and managed process improvement process that has been institutionalized within the system owner organization?
- ☑ Is there a policy for process improvement within the system owner organization?
- ☑ Have resources been allocated to the process improvement activity?
- ☑ Is there training for process improvement within the system owner organization?
- ☑ Are the assessments done in an objective manner?
- ☑ Were the assessment results documented and used to update existing processes?
- ☑ Were early phase assessments done fully and timely and used as input to the future phases to confirm moving in the correct direction?

Are there any other recommendations that can help?

Establish a documented system /software project management process. Before process improvement can be made, an initial process must be established as a baseline.

Tools will help establish and assess the system owner and development teams capabilities. The Software Institute at Carnegie Mellon University has established the Integrated Capabilities Maturity Model (CMMI). This model identifies the capabilities needed and the assessment tools used to rate an organization on its capability to perform systems/software development and project management. There are two models that have been established: 1) is the continuous model and 2) the staged model. The continuous model establishes profiles in each of the process areas and provides this profile to organizations after the assessment. The staged model provides a rating from level 0 (processes not documented or performed) to level 5 (processes are continuously optimized). Level 2 is considered the minimum level which an organization should have in place. For further information (see http://www.sei.cmu.edu/cmmi).

A closer look at: 1) establishing a documented systems engineering process and 2) assessment of the development team' capability to perform systems engineering.

For system owner or development team organizations that have not yet established a systems engineering process, this guidebook can provide a starting point for the establishment of the documented systems engineering process within the organization. This guidebook can serve as a tool to issue a request for qualifications to potential development teams to provide systems development services. It should be noted that the processes in this guidebook must be tailored to the goals and policies of the organization and, like any of the process standards, this is a guidance document, not an in-depth systems engineering or capabilities assessment document. It is recommended that each of the tasks identified be reviewed based on the size and complexity of the project and quality of products required.

For organizations that have documented processes and that advertise that they practice systems or software engineering in accordance with industry best practices, the level of maturity on how well they have preformed systems, software and project management would be of interest. For an organization, assessment should be done to determine how well they perform systems or software engineering. Assessments come in many forms, from internal micro-assessments to independent assessments done by a consultant. Currently there is no way to verify an internal assessment. The greatest confidence is when an organization has an external independent assessment performed on the organization.

A cautionary note: for large organizations, the level of capability for one part of the organization does not mean that the whole organization meets that level of maturity. For example, if a large company has many divisions, groups or business units, make sure that the advertised maturity level applies to the development team that will perform on your project.

It is also fair to request that the development team prove their documented processes for your review as a system owner and show how their documented processes map into the Capabilities Maturity Model.

4.8.8 Control Gates

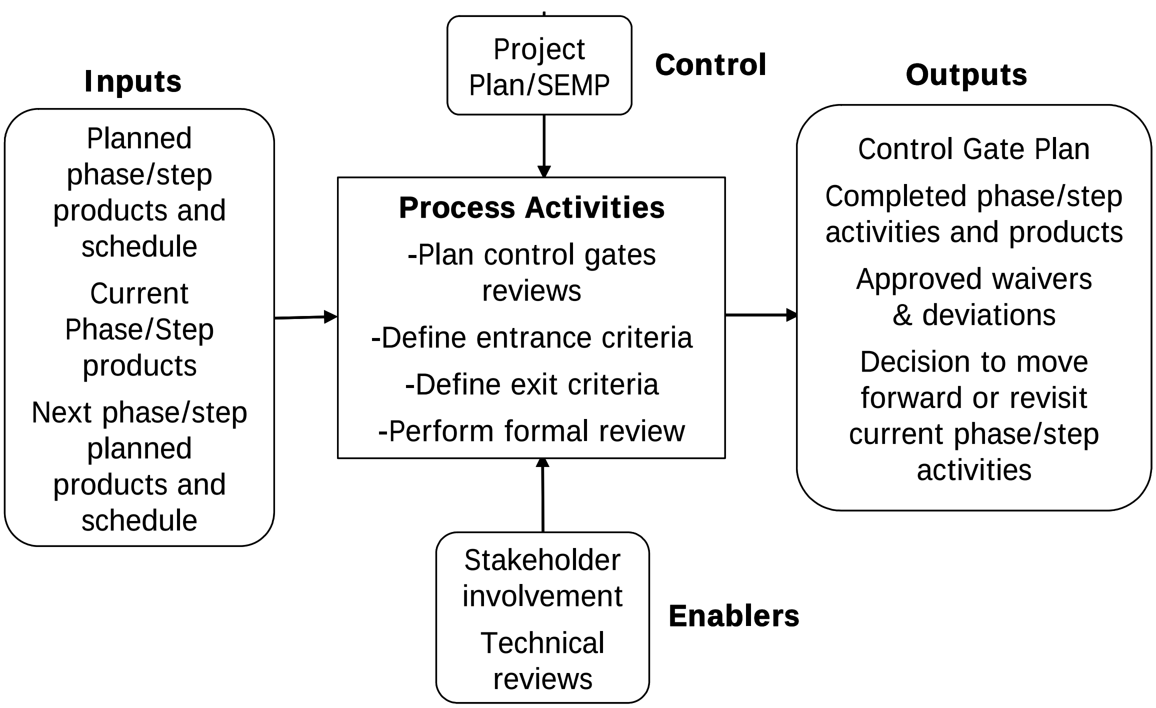
OBJECTIVE:

Control Gates define major decision points that are used to move from one phase of the project to the next. A control gate is used to determine if the products for the current phase of work are completed based on the criteria set out at the beginning of the project and that the project is ready to move forward to the next phase. Controls are used to get formal sign off of that phase of work by the system owner and management.

DESCRIPTION:

Control gates are used as a formal way to conclude and accept the products for a particular phase of the project. Intelligent Transportation Systems development, as laid out in this guidebook, has 6 major phases to the system lifecycle. Each phase has a major control gate and there are several additional control gates that occur within phases 1, 2, and 3 of the system lifecycle. These additional control gates are needed during the definition, development, integration, verification and deployment of the system. The additional control gates are used to evaluate the body of development work done for that step in the process and to determine if the project is ready (staff, funding, documentation, and products) to start the next step in the process. It is important that the control gates are planned in advanced as to how it will be conducted, who should be involved, how formal the control gate process should be, and what will be the entrance (what needs to happen before a control gate review takes place) and exit criteria (what conditions must be met before the next phase or step begins). Control gates are points at which the system owner has formally approved the completion of work for the current phase and has approved the team to move forward to the next phase. This approval is in the form of a **written sign-off of the phase of work** and a notice to proceed for the next phase.

CONTEXT OF PROCESS:



CONTROL GATES PROCESS

Inputs:

Planned Phase/Step Products and Schedule that were to be produced during this phase/step.

Current Phase/Step Products that are produced or developed during the current phase/step

Next Phase/Step Products and Schedule defines the plan and list of planned products that are to be developed for the next phase/step of work. (If this is a termination point for the effort of a team then this may be an internal plan for the next contract or an effort for a different organization)

Control:

SEMP/Project Plan that defines the control gate process and criteria for approvals.

Enablers:

Technical reviews identify the process for conducting the technical review.

Stakeholder involvement is essential to come to agreement for the completion of work for this phase and in agreement on the plans to move the project forward to the next phase/step.

Outputs:

Control Gate Plan is placed in the SEMP and will determine the process and criteria for performing a control gate.

Completion of all phase/steps activities and products all phase/step activities and products should have been completed for this phase/step of work.

Approved waivers and deviations any anomalies that have occurred but does not prevent the team from moving to the next phase of work needs to documented and approved by the system owner before proceeding to the next phase..

Decision to move forward or to revisit a current phase/step activity is made at this time. It may be that some of the phase work is not ready to move forward and some of the products need to be reworked or work from this current phase is better done in the next phase. These decisions are made and the plans are updated to reflect these decisions. Also, the team needs to show that they have the needed resources to move forward.

Process Activities:

Plan control gates reviews

Planning how the control gates will be performed includes who should attend, what are the entrance and exit criteria, and how formal the control gate needs to be.

Define entrance criteria

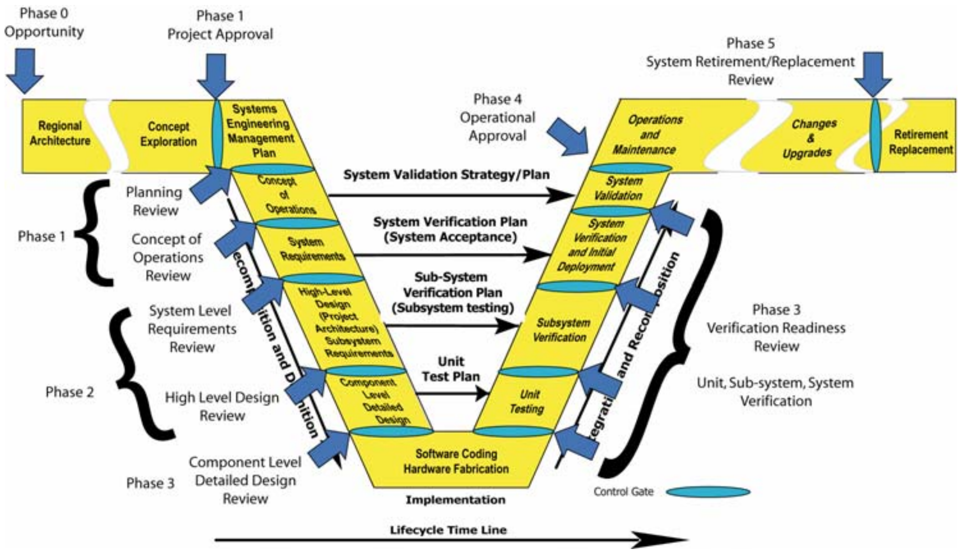
Before performing a control gate activity, the phase/step products and schedule should be reviewed and presentations developed in accordance with the plan. The purpose is to demonstrate that the phase of work has been completed. This may include, for example, the approved requirements document and verification plans, acceptance test results and the completion of all action items. If there is unfinished work and its completion needs to be moved to the next phase/step this needs to be identified with supporting rationale. Deviations or waivers should be issued on defective or incomplete work. Deviations allow the work to be used as is permanently and waivers allow the work to be used on a temporary basis until the defective/incomplete work is corrected.

Define exit criteria

If this is a continuation of effort, then a plan for the next phase/step of work is presented to ensure that the schedule, list of deliverables and resources are updated and in place in order to move forward to the next phase/step. If there is any dependency on work done in the current phase, this work is reviewed to ensure that it will support the next phase/step. If, at this control gate, the current effort is at an end or there is a change in who will be performing the next phase of work, this will provide a clear point of departure. For example, if the regional architecture work is completed and now projects are being implemented (control gate at phase 1), there may be a different system owner as well as a different development team. After the control gate at phase 3 there may be a different development team brought in for hardware/software development.

Perform formal review

Performing the control gate review should be done in accordance with the process developed for performing a technical review (see Technical Reviews).



Where do Control Gates take place in the project timeline?

Is there a policy or standard that talks about Control Gates?

FHWA Final Rule does not specifically mention general control gate practices. IEEE 1028-1988 Standards for Reviews and Audits provides information that is useful to control gates.

Which activities are critical for the system owner to do?

- Lead the planning of the control gate activities including the entrance and exit criteria.
- Lead the control gate reviews and gain stakeholder support for decisions made.
- Identify and gain stakeholder support and participation in the control gate activities.
- Lead the follow-up on any action items as a result of the control gate, including updating any plans, schedules, deliverables, waivers and deviations to the work.

How do I fit these activities to my project? (Tailoring)

Project size and number of stakeholders are the driving factors for this activity. On small projects where the system owner may be performing the control gate activities alone, in this case the control gate can be very informal and meeting minutes may be the only documentation needed to move the project forward. In multi-regional systems where there are a large number of stakeholders, the control gate activities will not be as simple, especially when consensus is sought for all decisions. A more formal and planned control gate will be needed where all of the stakeholders are involved with the planning activities and setting the criteria.

What should I track in this process step to reduce project risks and get what is expected? (Metrics)

On the technical side:

- Track the technical objectives for the phase/step and compare them with the planned technical objectives. Do the technical objectives meet the planned objectives? Were the correct technical objectives achieved?
- Incomplete set of technical objectives or a different set of technical objectives leads to increased technical risk and increased cost.
- At the control gate, is all documentation complete and/or does the documentation match the products required for the phase/step? Deficiencies in documentation will lead to reengineering portions of the product later on.

On the project management side:

- Track all products for the phase of work against the plan.
- Track deficiencies and their impact on the next phase of work. The next phase can be started even with deficiencies in the current phase of work. If there are deficiencies then the appropriate deviations or waivers must be issued.

Are all the bases covered? (Checklist)

- ☑ Is the plan for the control gate review included in the SEMP?
- ☑ Does it include entrance criterion?
- ☑ Does it include an exit criterion?
- ☑ Does it identify who should attend?

- ☑ Does it identify how the control gate will be conducted? (formally or informally)
- ☑ Have the entrance criteria been met prior to the control gate review?
- ☑ Have all phase/step products been reviewed and approved?
- ☑ Have the exit criteria been met for the next phase/step.
- ☑ Have all waivers and deviations been issued if any?
- ☑ Have the appropriate stakeholders been invited to the Control Gate review?
- ☑ Has the Control Gate review been conducted in accordance with the technical review process?

Are there any other recommendations that can help?

Have the appropriate stakeholders involved in the control gate review.

Frequent changes in stakeholders can be an obstacle to moving a project forward. If new stakeholders become involved mid-stream and they have not been completely updated on the project, they can cause "old" ground be covered again. When it comes to a control gate it is not the time to train new stakeholders on a project.

A closer look at the lifecycle control gate for ITS systems

Opportunity Control Gate (Phase 0)

The opportunity is reviewed and the stakeholders determine that this project has benefit to the region. An analysis is needed to identify the cost benefits to see if there is a good business case for the project.

Project Approval Control Gate: (Phase 1)

Out of the regional architecture a number of projects were proposed and a feasibility analysis was performed to provide a business case. This control gate approves projects to move forward to development and implementation.

Planning Control gate: (Phase 1)

Planning control gate reviews the Systems Engineering Management Plan (SEMP) to see if all the plans are complete enough to start the project.

Concept of Operations Control Gate (Phase 2)

Upon completion of the Concept of Operations, the control gate is used to see if the SEMP and Concept of Operations are complete and the stakeholders are in agreement on how the system is to work. This control gate starts the activities to define what the system is to do.

Requirements Control Gate (Phase 2)

This gate approves the system level requirements and the verification plans that will be used by the development team to implement the system.

High level Design Control Gate (Phase 3)

Here the high level design for the system is approved and is ready for the development team to start the component level detailed design. Sometimes this is called a Preliminary Design Review (PDR).

Component Level Detailed Design Control Gate (Phase 3)

This is the completion of the detailed design and the project is now ready for hardware/software development and purchase of COTS products. This is sometimes called the Critical Design Review (CDR)

Test Readiness Review Control Gates (phase 3)

This is a series of control gates that review the readiness of products from the development team to be verified, starting at the lowest level products and working up to sub-systems and finally to system acceptance.

Operational Control Gate (Phase 4)

This control gate approves the system to be commissioned into operation and maintenance.

System Retirement/Replacement Control Gate (Phase 5)

This control gate approves the system to be retired or replaced in part or in whole.

4.8.9 Trade Studies

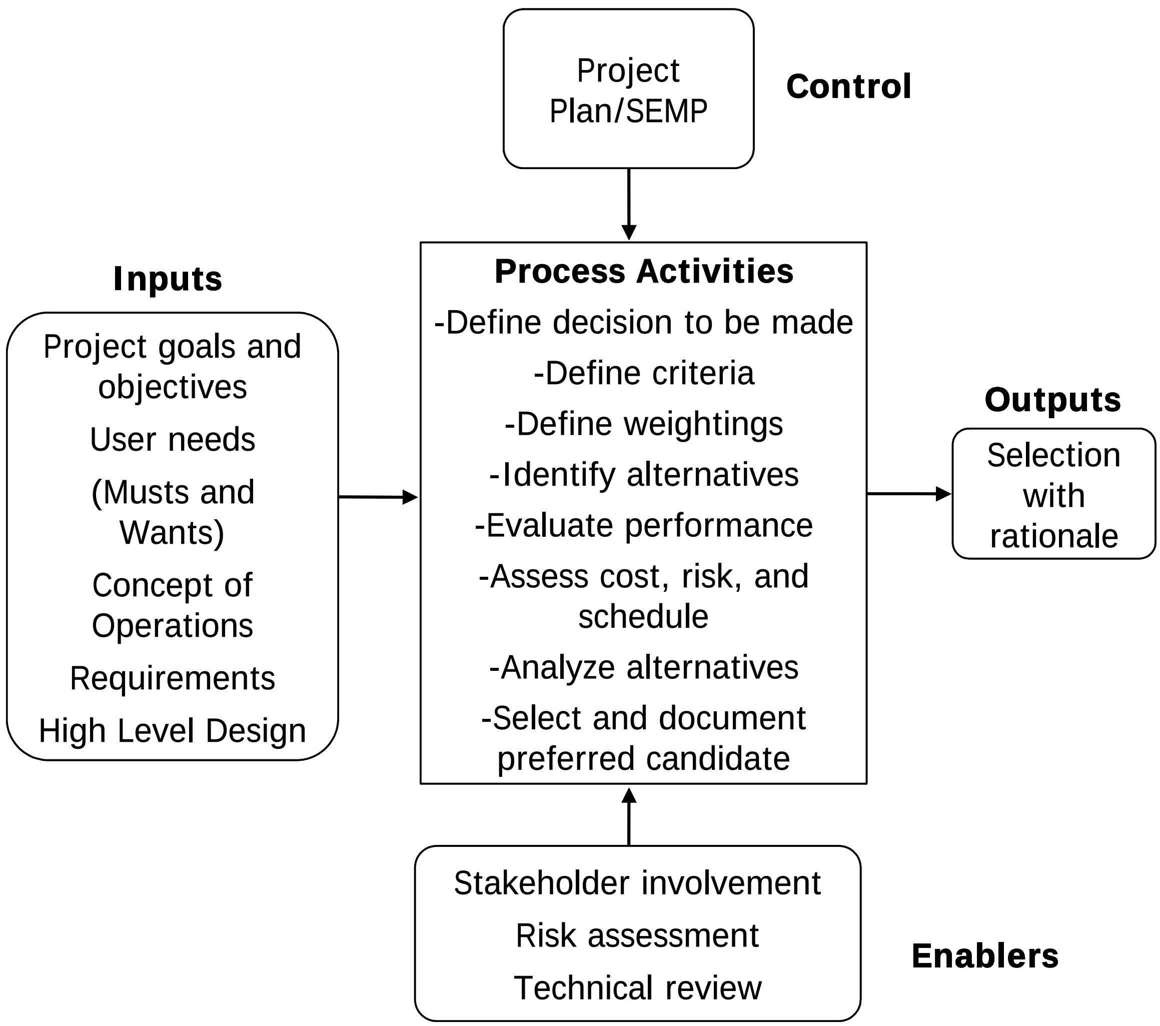
OBJECTIVE:

Trade studies compare the relative merits of alternative approaches, and so ensure that the most cost-effective system is developed. They maintain traceability of design decisions back to fundamental requirements. Trade studies do this by comparing alternatives at various levels for the system being developed. They may be applied to concept, design, implementation, verification, support, and other areas. They provide a documented, analytical rationale for choices made in system development.

DESCRIPTION:

Trade studies can be used in various phases and at different depths throughout the project to select from alternatives or to understand the impact of a decision. The inputs vary depending on what is being analyzed. For example, in concept exploration, the alternatives will be concepts, while in the design phase they will be design alternatives. The stakeholders are essential here to define and rate the criteria and to validate the results. The analysis may be done qualitatively, or by a model or simulation.

CONTEXT OF PROCESS:



TRADE STUDIES PROCESS

Inputs:

These inputs will be used only as available.

Project Goals and Objectives drive the selection of alternatives for concepts.

User needs and *Concept of Operations* drive the selection of alternatives for requirements.

Requirements and *High Level Design* drive the selection of alternatives for design elements.

Control:

SEMP and *Project Plan* constrain what may be developed and define budget and schedule.

Enablers:

Stakeholder involvement provides the key metrics and may suggest alternatives

Risk assessment evaluates each alternative relative to risk, balanced against effectiveness.

Technical reviews present the results and gather inputs and feedback.

Outputs:

Selection of the best of the alternatives, whether for concept, requirements, design or implementation, provides a choice based on solid analysis.

Rationale is the documentation of the alternatives compared, the criteria for selection, the analysis methodology, and the conclusions.

Process Activities:

Define decision to be made

First define the question that the trade study is to answer. This may be the selection of the most cost-effective concept or design. It may be to narrow down choices for more detailed evaluation. It may be to demonstrate that the choice made is the best one.

Define criteria

Experienced specialists will draw from the available inputs to identify the key evaluation criteria for the decision under consideration. These are measures of effectiveness, metrics that compare how well alternatives meet the needs and requirements. Examples are capacity (vehicles per hour), response time, throughput, and expandability.

Define weightings

Generally, there are multiple criteria, and so these same experts will assign each of them a relative weighting for relative importance.

Identify alternatives

Usually the trade study starts with alternatives, such as concepts or designs, to be compared or a specific question to be answered. Be sure that all reasonable alternatives are on the table.

Evaluate performance

Generally, the emphasis is on performance criteria, such as speed or effectiveness. For each alternative, the criteria may be evaluated quantitatively or qualitatively, and by such methods as simulation, performance data gathered from similar systems, surveys, and engineering judgment. These disparate evaluations are merged using the weighting factors to give a measure of overall effectiveness for each choice.

Assess cost, risk and schedule

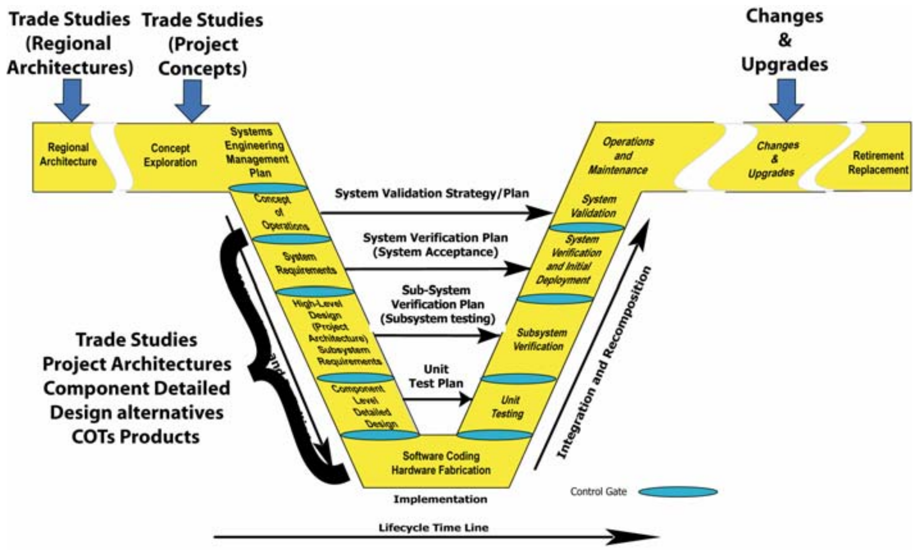
Estimate the cost of each alternative, not only the development cost, but the lifecycle cost, which includes operation and maintenance. Use the techniques of risk assessment (4.8.4) to compare the alternatives relative to technical or project risk. Determine the impact of each alternative on the schedule. Eliminate those that introduce too much risk of missing deadlines.

Analyze alternatives

Sensitivity analysis may also be used, especially with simulation, to see the effect of changes in subsystem parameters on overall system performance. The sensitivity analysis and the evaluations may suggest other, better alternatives.

Select and document the preferred candidate

Plotting overall effectiveness, based on the combined weighted metrics, against cost, or the other factors, is useful for picturing the relative merits and costs of each of the choices, and supports a good decision. Document the decision and the rationale behind it, to provide traceability back to the higher-level requirements. This document is also a repository of alternatives, in case a change is needed down the road.



Where do trade studies take place in the project timeline?

Is there a policy or standard that talks about Trade Studies?

FHWA Final Rule (23 CFR 940.11) requires the analysis of system configurations to meet requirements.

Which activities are critical for the system owner to do?

- Ensure that the proper stakeholders are involved.
- Suggest or elaborate on decision criteria
- Review the process and results of the trade studies.

How do I fit these activities to my project? (Tailoring)

The level of each activity should be appropriately scaled to the size of the project and the importance of the issue being traded off. For example, a small project will use qualitative measures and compare a small number of alternatives and do no sensitivity analysis. For example, an upgrade to a signal system will trade off features based on stakeholder priorities. A large project may use simulation to analyze key issues and perform sensitivity analysis.

What should I track in this process step to reduce project risks and get what is expected? (Metrics)

On the technical side:

These metrics check whether the set of alternatives is possibly driving a risky solution

- Number of high-risk alternatives selected
- Number of high-cost alternatives selected
- Number of selected alternatives that introduce schedule risk

On the Project management side:

- Percentage of alternatives examined
- Percentage of planned sensitivity analyses completed.

Are all the bases covered? (Checklist)

☑ Has a broad and reasonable selection of alternatives been examined?

☑ Does the rationale for the trade study conclusions flow out of the needs and requirements?

☑ Is the sensitivity of system effectiveness to changes in key parameters well understood?

☑ Is the selection rationale documented?

Are there any other recommendations that can help?

Trade studies should make maximum use of any ***previous work***, but if nothing applicable is available it will need to include more technical analysis. Often the two methods are combined--using analysis to predict system performance based on that of other systems. For example, well-documented improvements in traffic flow experienced when another agency implemented ramp metering could be combined with local data to predict the potential impact of a local ramp metering system.

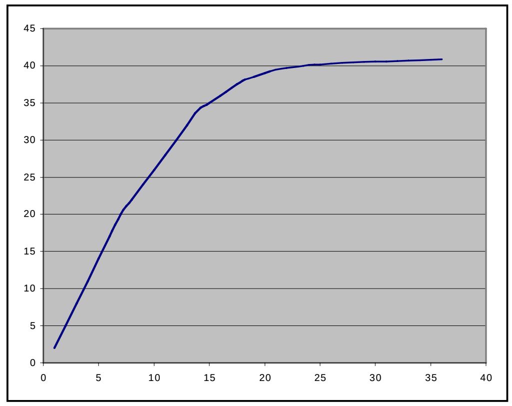
Simulation and modeling are tools that provide an objective, quantitative comparison of the merits of the alternatives. They may, for example, predict the effectiveness of each alternative in an operational scenario. These can range from a simple spreadsheet to a full traffic simulation.

A closer look at - Combining metrics There are usually multiple metrics for evaluating the system based on the various needs that the system is to meet. Generally, they are a mix of positive metrics (more is better, such as highway capacity) and negative metrics (less is better, such as response time). They also include both quantitative (e.g., predicted vehicle hours of delay) and qualitative (rating from 1 to 10). The units are varied—vehicles per lane per hour, seconds (of workstation response time), minutes (of incident response time), number (of predicted fatalities), % (of time available), and so on. It requires care to combine these into some measure of overall system goodness, without giving undue weight to one or the other. Section 4.8.5 gives a method for doing this.

Making qualitative measures quantitative Often time and available information do not allow a direct quantitative assessment. For example, the design of a regional Advanced Transportation Information System (ATIS) needed to focus on the key information needs of a large number of agencies in the region. There was very little time to do this prioritization, but there were dozens of documents that the agencies had produced discussing their needs. The approach used was to draw out from these documents any needs cited. Some agencies listed their "top ten" information needs in rank order. These were assigned 1 to 10 points, depending on their place in the list, 10 being best. If a need was cited without ranking it relative to other needs, it was given a medium rating of 5 points. The total points for any need were then a metric indicating how many agencies needed the particular information and how strongly they felt about it.

If you hold a workshop to collect stakeholders' preferences, here is a simple way to get their inputs on alternatives. First discuss the alternatives and their pros and cons. Then list them on a flipchart and give each participant a few colored adhesive dots. Be sure each participant gets the same number of dots, about 10 – 20% of the number of alternatives. Allow each participant to place their dots next to the choices that he favors, even placing multiple dots against a choice that he particularly likes. The number of dots is a metric for stakeholder preference. This type of metric could be used to compare alternatives directly or to determine relative weights for multiple metrics.

Sensitivity analysis Using a simulation or other analytical tool lets you vary design parameters over their potential values and predict the effect on performance. The "knee of the curve" shows where more stringent design requirements give little system improvement.



In the example chart, the knee occurs around 15 to 20 for the design parameter (horizontal axis). There is very little performance improvement (vertical axis) from a more stringent design. Sensitivity analysis can also be done in multiple dimensions to determine, for example, whether money should be spent on improving communications or detectors.

Estimating costs for alternatives There is an art to predicting the cost of a new system. A lifecycle cost analyst can do it by extrapolating from existing systems. Qualitative assessments are often sufficient. Examples are high/medium/low, in cost or difficulty to implement. Plotting effectiveness versus cost would support the decision.

4.8.10 Technical Review

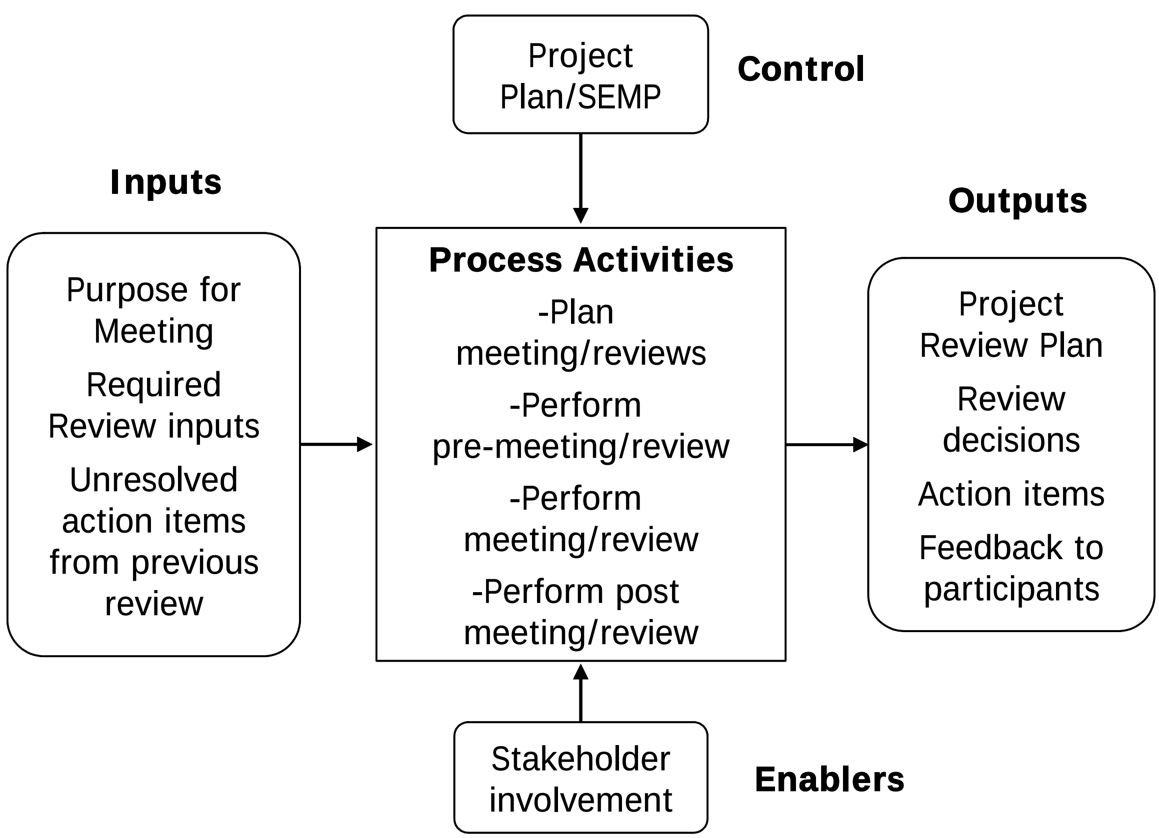
OBJECTIVE:

Technical reviews provide a structured and organized approach to reviewing project products to determine if they are fit for their intended use. This section also describes a process to plan and conduct a meeting that can be used for the different types of technical reviews. Technical reviews are used to identify design defects, suggest alternative approaches, communicate status, monitor risk, and coordinate activities within multi-disciplinary teams.

DESCRIPTION:

Technical reviews are critical to the success of Intelligent Transportation System projects. Technical reviews provide status and feedback on the products under review and on on-going activities of a project. A technical review is the primary method for communicating progress, coordinating tasks, monitoring risk, and transferring products and knowledge between team members of a project. The Institute of Electrical and Electronics Engineers (IEEE) 1028-1998 standard on software reviews and audits identifies the following five types of reviews: management reviews (for example, control gates), technical reviews, inspections (primarily for identifying errors or deviations from standards and specifications), walk-throughs (for example, requirement or design walk-throughs) and audits (for example, physical and functional audits used as part of the configuration management process). The process for conducting review meetings should be established in the Project Plan/SEMP and be carried out the same way for each review. The differences in reviews would be in content and the level of formality. This formality would be tailored for the type of review and its purpose. This section describes a basic meeting procedure including pre-meeting activities, conduct, and post meeting activities.

CONTEXT OF PROCESS:



TECHNICAL REVIEW PROCESS

Inputs:

Purpose for the meeting must be clearly established with expected outcomes.

Required Review Inputs should be provided. These are products for the phase under technical review.

Unresolved action items from previous review are carried over for continuing discussion and/or decisions.

Control:

Project Plan/SEMP contains the process that will be used to perform technical reviews.

Enablers:

Stakeholder involvement is needed for stakeholders to participate in and to fill various roles for the technical review process.

Outputs:

Project Review Plan will identify how technical reviews will be carried out for the project. This will be part of the Project Plan/SEMP.

Review of decisions includes the documented acceptance, re-work with comments, and deviations and waivers to the phase products by the participants of the technical review.

Action items are assigned with completion dates. Critical items are tracked between meetings if necessary.

Assignments are documented and sent out as part of the feedback to the participants. This feedback should have a definition of the action item and a planned date for completion.

Feedback to participants documents the results of the meeting and provides a record of the meeting back to participants for their review and comments. This ensures that decisions, actions and assignments were accurately documented.

Process Activities:

Plan reviews

A plan is developed for the technical reviews for a project. This plan includes the schedule for reviews, who (functionally) will be in attendance, level of formality for each review, entry criteria (drafts, final products) what will be the process for the review (structured presentation or informal round-table), and what will be the exit criteria (100% consensus agreement, majority agreement, project manager only).

Perform pre-meeting (review) activities:

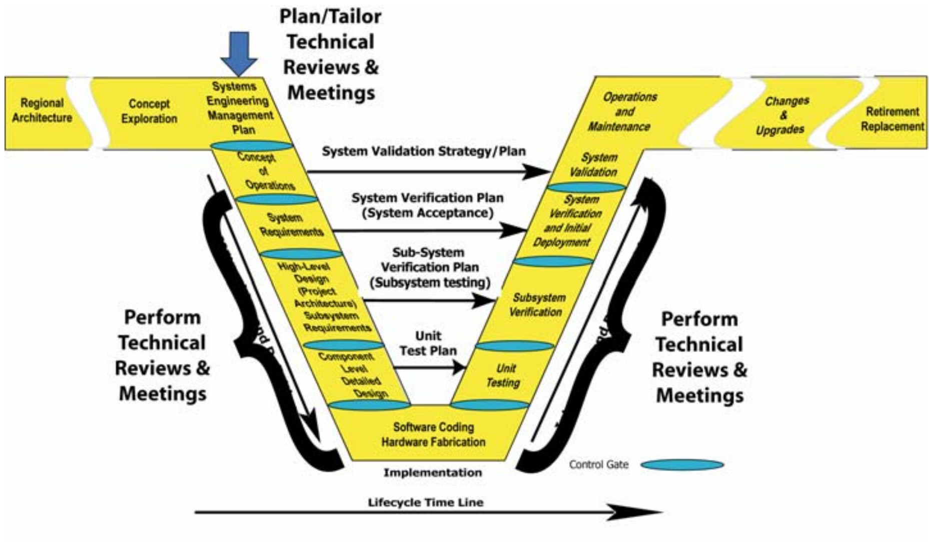
Define the purpose, objectives, and the intended outcomes of the meeting. Prepare an agenda, identifying participants and their roles and distribute the agenda and background material. Reserve and inspect the meeting facilities and location to see if all needed equipment is in working order and that the facility meets the needs for the meeting. Example items to look for are space (size and shape), break rooms, rest rooms, lunch facilities, break-out rooms, climate control, lighting, noise levels, appropriate furniture and configuration, equipment and electrical. Make arrangements, if necessary, for breakfast, lunch, dinner and/or break refreshments.

Perform meeting/review

Technical meetings should start and end on time. The purpose of the meeting should be clearly stated, an updated agenda provided to the attendees, and a roster that documents the attendees that are present with up to date contact information (email, phone number, organization) and their role (e.g. presenter, chairman, or observer) in the meeting should be placed with the meeting minutes. The ground rules for the meeting should be reviewed prior to discussion. Starting with unresolved action items from the previous review, conclude one agenda item at a time, manage discussions so that there is focus on the topic, follow the pre-arranged ground rules, keep track of the time and document all decisions, actions and assignments. At the close of the meeting summarize all decisions, actions and assignments, review agenda items and assignments for the next meeting, confirm date, time and place of the next meeting and finally end on time.

Perform post meeting/review activities

The meeting should be followed up with a complete set of minutes that include all decisions, actions, and assignments. The minute taker, if needed, should follow up with the attendees to make sure the minutes are as complete as possible. These minutes and any supporting material should distributed back to the attendees promptly for review and comment. Assignments should be completed, and progress checked on critical assignments. Honor commitments for the next meeting. Carry over unresolved actions with status and recommended resolutions.



Where do Technical Reviews take place in the project timeline? Is there a policy or standard that talks about Technical Reviews?

FHWA Final Rule does not specifically mention general reviews and meeting practices. IEEE 1028-1988 Standards for Reviews and Audits provides information that is useful to control gates.

Which activities are critical for the system owner to do?

- Lead the definition and documentation of the process in conducting a technical review.
- Gain stakeholder support in the participation of technical reviews.
- Lead the participation of technical reviews.
- Review decisions, actions and assignments from the technical review.
- Follow-up on critical assignments.

How do I fit these activities to my project? (Tailoring)

In this activity, the number of reviews and level of formality is tailored to the size and type of the project. For example, on a small traffic signal control system that is a COTS product, the number of reviews can be minimal (bi-weekly or monthly). The meetings may be informal with project manager and/or traffic engineer in a review of progress. The feedback may be just a summary of the meeting minutes.

What should I track in this process step to reduce project risks and get what is expected? (Metrics)

Technical and project management:

Technical reviews are used to identify design defects, suggest alternative approaches, communicate status, monitor risk, and coordinate activities within multi-disciplinary teams. This would be the time and place to monitor, review and take action on both technical and project management metrics that were set up for the phase that is currently in progress.

Are all the bases covered? (Checklist)

- ☑ Was a review plan developed for the project?

  Did the plan contain:
  - ☑ The number or frequency of the reviews?
  - ☑ The process for carrying out each review?
  - ☑ Roles identified for each review?
  - ☑ Level of formality identified for each review?
- ☑ Was a technical review agenda developed and distributed well ahead of the scheduled meeting date?
- ☑ Was all the supporting and background material generated and distributed to the attendees well ahead of the scheduled meeting date (per the plan)?
- ☑ Were the attendees and their role identified or defined (per the plan)?
- ☑ Were the time and location identified?
- ☑ Were the purpose and outcomes identified?
- ☑ Were all unresolved assignments, identified in the previous meeting,

brought forward to the upcoming meeting?

- ☑ Has the location of the technical review been checked out for size, climate, configuration, equipment, furniture, noise, and lighting?
- ☑ Are all the presenters well prepared for the meeting?
- ☑ Were the ground rules for the meeting discussed before the start of discussion?
- ☑ Did the meeting start on time?
- ☑ Were introductions made by all attendees?
- ☑ Was an attendance roster created for the meeting with up to date contact information for each attendee?
- ☑ Was the purpose of the meeting clearly stated and what are the expected outcomes?
- ☑ Was an updated agenda provided for each attendee with the priorities assigned for each agenda item?
- ☑ Was each agenda item concluded before discussing the next or other items?
- ☑ Were all decisions, actions and assignments documented as part of the minutes and summarized at the end of the meeting?
- ☑ Did the meeting end on time?
- ☑ Were the minutes distributed to the attendees?
- ☑ Were all critical assignments followed up between meetings?

Are there any other recommendations that can help?

Have ground rules for technical reviews. The following is a recommended set of ground rules that participants observe during the meeting:

- We tell it like it is but respect, honor, and trust one another.
- We work toward consensus recognizing that disagreements in the meeting are OK – but once we agree, we all support the decision.
- We have one conversation at a time and our silence is consent.
- We focus on issues, not on personalities and we actively listen and question to understand.
- We do not attack the messenger.
- We start on time, observe time limits, and structure the agenda to end on time.

A closer look at the types of technical reviews used throughout the project timeline

***Planning** review is to verify that all plans appropriate for the project are identified*. Tailoring for each plan is reviewed and updated as needed.

***Concept of Operations** review is to ensure that the operation of the system being defined is appropriate and to addresses the needs of the stakeholders*. This is a critical review, as the concept of operations will identify the operational needs, needed agreements, candidate external interfaces, and maintenance responsibilities.

***Requirements** review is used to ensure that the system and sub-system requirements and verification plans are appropriate for the system being defined.* This review verifies that the requirements are complete and that each meets all the criteria for a good requirement and traces to user needs.

***High level Design** review is to ensure that the project level architecture is well formed, balanced and appropriate for the problem space and that the functionality and performance of the defined system meet the intended need*. This review verifies that the project architecture is consistent with the regional architecture and, if necessary, documents the differences. This is a major technical review and is sometimes called a PDR (Preliminary Design Review).

***Component level detailed design** review is used to ensure that the detailed design is ready for implementation.* This is a major review since when completed the detailed design is ready for implementation. This is sometimes called the CDR (Critical Design Review).

***Test Readiness** review is used to see if the components, sub-systems and system are ready for verification.* For each level of verification there should be a review prior to the formal verification of the product.

***Operational** Review is used to ensure that the system is ready for deployment.* This review verifies that all training and support for the system is in place and that the operations and maintenance personnel are ready.

5 Roles and Responsibilities in Systems Development

OBJECTIVE:

This section describes the various roles that will be performed in the development lifecycle of an ITS project. It will provide guidance on what the roles entail and who are the potential candidates to perform them. A matrix shows the role that the system owner, the systems engineering assistant, and the development team have in each stage of the development lifecycle.

The role of the system owner, systems engineering assistant and development team will vary in level of involvement as well as areas of responsibility throughout the project lifecycle. This section will provide guidance in each development step and identify the roles that are needed in each phase of the lifecycle of an ITS project. Following this discussion, a matrix of roles and responsibilities is provided.

System Owner (Project Sponsor & Stakeholders)

The system owner, or project sponsor and stakeholders, who is implementing an ITS will need to acquire a set of development services, either in-house or contracted, to develop the ITS project. The system owner and operating organization will ultimately be responsible for the system and its operations and maintenance. The system owner will need to supply clear requirements and expected outcomes for the project to the development team. These outputs must be compatible with the long-term operations and maintenance goals of the system owner and the stakeholders. The success of the project will rely on the system owner having a good working relationship with the systems engineering assistant and development team(s) that implements the system. This section will identify the roles and responsibilities of the system owner at each phase of the ITS lifecycle from the interface to the regional ITS architecture to retirement/replacement of the system or major system elements.

Systems Engineering Assistant (In-house staff, Independent Verification and Validation, System Manager)

The systems engineering assistant provides the system owner with specialty support in systems engineering. This role can be undertaken by in-house staff or a system manager, or by an Independent Verification and Validation consultant (IV&V) to a limited extent. Contract resources are particularly valuable for large, complex, or unusual projects, or if the system owner's organization is small and lacks systems engineering expertise, as may be the case with a small or medium size city or MPO. The systems engineering assistant defines requirements and the project architecture, prepares the request for proposal or other system procurement documents, assists in the review of proposals, provides independent review services (Independent Verification & Validation), and provides technical assistance to the system owner during the lifecycle of the ITS system. A consultant who neither offers products nor is affiliated with a development team or vendor can remain un-biased in the selection and evaluation of developers and products. Also, the consultant can assist the agency in Configuration Management, Risk Management, development team evaluations and in process improvement. It is important to find a consultant that has both systems engineering expertise and Intelligent Transportation Systems knowledge and experience. The role of the systems engineering assistant is described for each phase of the system lifecycle in the following matrix.

Development Team (In-house, Systems Integrator)

An ITS system development team normally develops or supplies hardware and software that integrate custom (project-specific) and off-the-shelf products. The system owner would secure the services of a development team to perform the detailed design, develop any necessary custom hardware or software, integrate off-the-shelf products, and verify the system as a whole. The development team may be another department within the system owner's organization (internal development teams), or a contracted integration team, which would be the normal case for most organizations. The role of the development team is described for each phase of the lifecycle.

The following Table 5-1 through Table 5-5 recommends the different roles and responsibilities during each phase of the project life cycle. The number associated with each phase or task refers to the corresponding section in this Guidebook.

Table 5-1 Phase 0 Roles and Responsibilities

Roles and Responsibilities
ITS Project Lifecycle

Phase & Task	4.2.1 Interfacing with Planning & the Regional ITS Architecture	4.2.2 Needs Assessment	4.2.3 Concept Selection & Feasibility Assessment
	Page 1 of 5 Phase 0 Concept Exploration & Feasibility Analysis		
	Roles and Responsibilities		
System Owner Project Sponsor & Stakeholders	✓ Coordinate with Planning ✓ Identify applicable regional ITS architectures ✓ Ensure that project goals and objectives are sufficiently clear to support subsequent tasks	✓ Provide initial statement of needs ✓ Provide current inventory ✓ Provide supporting documentation ✓ Identify initial set of stakeholders – encourage participation ✓ Actively participate in elicitation to gather needs	✓ Describe, refine and approve needs, vision, goals, objectives and constraints ✓ Review and approve criteria ✓ Review and approve candidate concepts ✓ Review and approve conclusions
Systems Engineering Technical Assistance In-house, Independent Verification and Validation[1] Consultant, Systems Manager	✓ Document the applicable regional ITS architecture ✓ Document the goals and objectives ✓ Document the relevant interface elements between the projects and planning ✓Inventory ✓Stakeholder Identification ✓High level Needs/Services ✓High level requirements ✓Area of coverage ✓Operational Concept ✓Interconnect/Information flows ✓ITS standards ✓Project sequencing ✓Agency agreements ✓Constraints ✓ Document consistency	✓ Refine and update Stakeholder lists ✓ Facilitate the elicitation process ✓ Document, validate and prioritize needs ✓ Perform a gap analysis ✓ Document the problem space ✓ Identify and document key resources ✓ Assess needs against cost and effect on the problem space	✓ Document vision, goals and objectives ✓ Refine and update constraints ✓ Develop evaluation criteria ✓ Identify candidate alternative solutions ✓ Synthesize alternative project concepts ✓ Compare and evaluate alternative concepts ✓ Document alternatives, recommendations, and rationale for each ✓ Perform a cost/benefit analysis for each concept (feasibility analysis) ✓ Recommend a concept
Development Team In-house, System Integrator	✓ Review and comment* *at times the architecture is open to public comment e.g. Industry review of architecture	✓ Review and comment* *at times the needs are open to public comment e.g. Industry review of needs	✓ Review and comment* *at times the concept, vision, goals and objectives are open to public comment e.g. Industry review of Vision, Goals and Objectives

[1] Independent Verification and Validation - role and responsibility are monitoring, reporting, supporting and participating but not performing – applies in all phases and tasks.

Table 5-2 Phase 1 Roles and Responsibilities

Roles and Responsibilities
ITS Project Lifecycle

Phase & Task	4.3.1 Project Planning	4.3.2 Systems Engineering Management Planning	4.3.3 Concept of Operations
	Page 2 of 5 **Phase 1 Project Planning & Concept of Operations Development**		
	Roles and Responsibilities		
System Owner Project Sponsor & Stakeholders	✓ Approve project tasks, plans, budgets and schedules ✓ Approve scope of project ✓ Approve supporting internal resources ✓ Approve management plans (Risk, QA, CM) ✓ Manage RFP and Procurement	✓ Establish framework of technical plans for the project ✓ Approve project products ✓ Approve control gate criteria ✓ Approve project organization ✓ Approve initial set of risks ✓ Establish Configuration Management process and board	✓ Review and approve project level vision, goals, objectives. ✓ Approve and review project level stakeholder list ✓ Participate in, and review and approve the Concept of Operations and other planned phase products ✓ Monitor and update risks

Systems Engineering Technical Assistance In-house, Independent Verification and Validation[1] Consultant, Systems Manager	✓ Identify, prepare, and document
	  ✓ Scope of project
	  ✓ Budgets, project tasks (SOW's) and schedules
	  ✓ Needed resources
	  ✓ Project plan
	  ✓ Management Plans
	  ✓ RFP, RFQ, RFI
	  ✓ Procurement documents
	  ✓ Evaluation criteria
		✓ Identify, prepare, and document
		  ✓ Technical plan framework
		  ✓ Technical SOW
		  ✓ Development strategy
		  ✓ Technical evaluations
		  ✓ Supporting plans
		  ✓ Tailoring options
		  ✓ Review and control gate process
		  ✓ Initial set of project risks and risk plan
			✓ Identify, prepare, and document
			  ✓ Project level vision, goals and objectives
			  ✓ Operational scenarios
			  ✓ Concept of operations
			  ✓ Validation strategy and plan
			  ✓ Project specific stakeholders
			  ✓ Update project risks
			✓ Support, participate & report
			  ✓ Control gates

Development Team In-house, System Integrator	✓ Review and comment*
	✓ Development schedules
	✓ Technical plans
	✓ Statement of Work
	
	*at times the project planning is open to public comment
	e.g. Industry review and comment
		✓ Review and comment*
		  ✓ Technical plan framework
		  ✓ Technical SOW
		  ✓ Development strategy
		  ✓ Technical evaluations
		  ✓ Supporting plans
		  ✓ Tailoring options
		  ✓ Review and control gate process
		  ✓ Initial set of project risks and risk plan
		*Internal development teams
			✓ Review and comment*
			  ✓ Project level vision goals and objectives
			  ✓ Operational scenarios
			  ✓ Concept of operations
			  ✓ Validation strategy and plan
			  ✓ Project specific stakeholders
			  ✓ Update project risks
			
			* Internal development teams

[1] Independent Verification and Validation - role and responsibility are monitoring, reporting, supporting and participating but not performing – applies in all phases and tasks.

Table 5-3 Phase 2 Roles and Responsibilities

Roles and Responsibilities
ITS Project Lifecycle

Phase & Task	4.4.1 Requirements Development	4.4.2 High Level Design	4.4.3 Component Detailed Design
	Page 3 of 5 — Phase 2 System Definition and Design		
	Roles and Responsibilities		
System Owner Project Sponsor & Stakeholders	✓ Encourage stakeholder participation ✓ Participate in the elicitation process to gather requirements ✓ Review and approve requirements baseline ✓ Approve system level requirements baseline ✓ Monitor the trends after baseline ✓ Monitor project risks	✓ Encourage stakeholder participation ✓ Negotiate and approve interface agreements ✓ Suggest and review alternative project architectures ✓ Review and approve project architecture ✓ Monitor the trends after baseline ✓ Monitor and update project risks ✓ Approve RFP for system Integrator	✓ Encourage stakeholder participation ✓ Participate in the technical reviews ✓ Review and approve detailed design (critical design review) ✓ Participate in the alternative COTS product review ✓ Monitor the trends after baseline ✓ Monitor and update project risks ✓ Contract for system integrator

Systems Engineering Technical Assistance In-house, Independent Verification and Validation[1] Consultant, Systems Manager	✓ Identify, prepare, and document (Update)
	  ✓ Validated set of system level requirements
	  ✓ Requirements reviews
	  ✓ System verification plan
	  ✓ System interfaces
	  ✓ Project risks and trends
	✓ Perform requirements
	  ✓ Elicitation
	  ✓ Analysis
	  ✓ Decomposition
	  ✓ Management (traceability, attributes, baseline, quality)
	  ✓ Completeness check
	✓ Perform/Support requirements
	  ✓ Feasibility analysis
	✓ Support, participate and report
	  ✓ Control gates
		✓ Identify, prepare, and document (Update)
		  ✓ System integrator RFP
		  ✓ Validated set of sub-system requirements
		  ✓ Requirements reviews
		  ✓ Project level architecture, interfaces and standards
		  ✓ Sub-system verification plans
		  ✓ Project risks and trends
		  ✓ Configuration items
		✓ Perform requirements
		  ✓ Elicitation
		  ✓ Analysis
		  ✓ Decomposition
		  ✓ Management (traceability, attributes, baseline, quality)
		  ✓ Completeness check
		✓ Perform/Support requirements
		  ✓ Feasibility analysis
		✓ Support, participate, and report
		  ✓ Control gates
			✓ Support, participate, review and comment:
			  ✓ System integrator evaluation
			  ✓ COTS evaluation
			  ✓ Detailed designs
			  ✓ Technical reviews
			  ✓ Developmental CM process
			  ✓ Vendor CM process
			  ✓ Developmental risk management
			  ✓ Unit level verification plan
			  ✓ Technical plans
			    ✓ Integration
			    ✓ Deployment
			    ✓ Installation
			    ✓ Technology (COTS evaluations)
			    ✓ Security
			    ✓ Development
			    ✓ Operations and maintenance
			  ✓ Support, participate, and report
			    ✓ Control gates

Development Team In-house, System Integrator	✓ Review and comment
	  ✓ Validated set of system level requirements
	  ✓ Requirements reviews
	  ✓ System verification plan
	  ✓ Project risks and trends
	✓ Support requirements
	  ✓ Elicitation, analysis, decomposition, management (traceability, attributes, baseline, quality), completeness check
	✓ Support/Perform requirements
	  ✓ Feasibility analysis
		✓ Review and comment
		  ✓ Requirements
		  ✓ Project level architecture, interfaces and standards
		  ✓ Sub-system verification plans
		  ✓ Project risks and trends
		  ✓ Configuration items
		✓ Support requirements
		  ✓ Elicitation, analysis, decomposition, management (traceability, attributes, baseline, quality), completeness check
		✓ Support/Perform requirements
		  ✓ Feasibility analysis
			✓ Identify, prepare and document technical plans per SEMP framework
			✓ Perform and document
			  ✓ Detailed designs and models
			  ✓ Technical reviews (CDR)
			  ✓ Developmental CM process
			  ✓ Developmental risk management

[1] Independent Verification and Validation - role and responsibility are monitoring, reporting, supporting and participating but not performing – applies in all phases and tasks.

Table 5-4 Phase 3 Roles and Responsibilities

Roles and Responsibilities ITS Project Lifecycle

Page 4 of 5	Phase 3 System Development & Implementation			
Phase & Task	4.5.1 Hardware / Software Development	4.5.2 Integration	4.5.3 Verification	4.5.4 Initial System Deployment
	Roles and Responsibilities			
System Owner Project Sponsor & Stakeholders	✓ Review and participate in technical reviews ✓ Review CM activities management process ✓ Monitor risks ✓ Review and encourage coordination between projects ✓ Procure COTS	✓ Review integration plan and performance ✓ Identify and obtain integration support ✓ Monitor integration definition and performance progress ✓ Monitor risk ✓ Train operational and maintenance staff	✓ Review verification master plan and performance ✓ Approve verification procedures ✓ Obtain verification support ✓ Witness verification ✓ Monitor the verification process ✓ Monitor risk	✓ Approve deployment ✓ Review deployment plan and performance ✓ Obtain deployment support ✓ Review operational readiness of staff ✓ Witness deployed system verification ✓ Accept system ✓ Monitor deployment process ✓ Monitor risk
Systems Engineering Technical Assistance In-house, Independent Verification and Validation[1] Consultant, Systems Manager	✓ Support, participate & report ✓ Code and hardware implementation reviews ✓ Control gates ✓ Monitor and report ✓ System and developmental risks and CM activities ✓ COTS procurement ✓ Perform and report ✓ Coordination between project developments ✓ Risk assessment ✓ Risk mitigation	✓ Support, participate & report ✓ Integration reviews ✓ Control gates ✓ Staff training ✓ Monitor and report ✓ System and integration risks and CM activities ✓ Perform and report ✓ Coordination between project integration ✓ Risk assessment ✓ Risk mitigation	✓ Support, participate & report ✓ Verification readiness reviews ✓ Control gates ✓ Verification procedures ✓ Configuration item, sub-system and system verification ✓ Monitor and report ✓ System and verification risks and CM activities ✓ Perform and report ✓ Coordination between project verification ✓ Risk assessment ✓ Risk mitigation	✓ Support, participate & report ✓ Deployment readiness reviews ✓ Control gates ✓ Deployed system verification and acceptance ✓ Operations & maintenance training ✓ Monitor and report ✓ System and deployment risks and CM activities ✓ Perform and report ✓ Coordination between project deployments ✓ Risk assessment ✓ Risk mitigation
Development Team In-house, System Integrator	✓ Perform and document ✓ Development of hardware and software components and interfaces, databases, communications, COTS applications ✓ Technical reviews ✓ Prototyping ✓ Risk identification, assessment and mitigation ✓ Unit verification plans ✓ Developmental configuration management ✓ Implement and document ✓ Development environment ✓ Participate and support CM, RM	✓ Perform and document ✓ Configuration item, sub-system and system integration ✓ Integration reviews ✓ Risk identification, assessment and mitigation ✓ Integration defect identification and resolution per the plan ✓ Implement and document ✓ Integration environment ✓ System user and maintenance documentation and staff training ✓ Participate and support ✓ System level CM, RM	✓ Develop and document ✓ Configuration item, sub-system and system level verification procedures per plan ✓ Perform and document ✓ Configuration item, sub-system and system verification ✓ Verification readiness reviews ✓ Risk identification, assessment and mitigation ✓ Verification defect identification and resolution per plan ✓ Implement and document ✓ Verification environment ✓ Participate and support ✓ System level CM, RM	✓ Perform and document ✓ Initial deployment ✓ Deployment reviews ✓ Risk identification, assessment and mitigation ✓ Deployment defect identification and resolution per the plan ✓ Deployed system, burn-in, evaluation, verification and acceptance ✓ Implement and document ✓ Deployment environment ✓ Operations and maintenance training ✓ Document and report ✓ Verification defects and resolution ✓ Participate and support ✓ System level CM, RM

[1] Independent Verification and Validation - role and responsibility are monitoring, reporting, supporting and participating but not performing – applies in all phases and tasks.

Table 5-5 Phase 4 & 5 Roles and Responsibilities

Roles and Responsibilities
ITS Project Lifecycle

Phase & Task	Page 5 of 5 Phase 4 Operations & Maintenance			Phase 5 System Retirement / Replacement
	4.6.1 System Validation	4.6.2 Operations & Maintenance	4.6.3 Changes & Upgrades (changes and upgrades are performed in accordance with 4.3-4.6)	4.7.1 System Retirement / Replacement
	Roles and Responsibilities			
System Owner Project Sponsor & Stakeholders	✓ Review validation strategy and plan and updates ✓ Obtain stakeholder support for validation ✓ Review pre-system baseline measurements as necessary (before and after measurements) ✓ Review validation results and recommendations ✓ Monitor risks	✓ Approval to commission system into operations and maintenance ✓ Review and approve updates to the operations and maintenance plan ✓ Define the on-going monitoring of system performance ✓ Initiate lifecycle CM activities for system ✓ Collect operations and maintenance information ✓ Monitor risks	✓ Elicit stakeholder involvement changes and upgrades ✓ Perform activities as defined in 4.3-4.6 for the system owner ✓ Approve reverse engineering activities if the legacy system is not documented in the area of change and upgrade ✓ Review changes and documentation updates (CM process) ✓ Assess cost/benefit of change (CM process) ✓ Monitor risks	✓ Elicit stakeholder involvement in retirement/replacement of system or major sub-systems ✓ Review and update the retirement replacement plan ✓ Approve the cost/benefit of current system ✓ Review and approve the cost/benefit of the changes and upgrade CM process ✓ Review RFI to industry for candidate replacements ✓ Monitor risks
Systems Engineering Technical Assistance In-house, Independent Verification and Validation[1] Consultant, Systems Manager	✓ Perform and report ✓ Validation strategy and plan updates ✓ Pre-system evaluation ✓ Perform post-system evaluation ✓ Assess system benefit ✓ Perform and document ✓ Systems analysis ✓ Strengths and weaknesses ✓ Requirements for next evolution	✓ Perform/support, or participate & report** ✓ Operational assessment ✓ Maintenance assessment ✓ Lifecycle CM activities	✓ Support, participate & report** ✓ Change review and assessment ✓ Change management ✓ Status accounting and audits ✓ Contracting for changes and upgrades ✓ Technology demonstrations ✓ Perform and document ✓ Reverse engineering activities ✓ Change and upgrade activities as defined in 4.3-4.6 for the systems engineering technical assistance	✓ Perform and document ✓ Replacement/upgrade assessment ✓ Perform gap analysis between legacy system capabilities and needed system capabilities ✓ Evaluation of cost/benefit for replacement ✓ Technology demonstration ✓ Develop and document ✓ Replacement strategy and plan ✓ RFI to industry for replacement options
Development Team In-house, System Integrator	✓ Support and review ✓ Validation of system ✓ Validation strategy and plan	✓ Perform and support ✓ Initial operations and maintenance ✓ Initial on-call service ✓ Long term support and maintenance	✓ Perform, document and support ✓ Change and upgrade activities as defined in 4.3-4.6 for the development team ✓ Technology demonstrations	✓ Perform and support ✓ Industry response to potential solutions ✓ Technology demonstrations ✓ Replacement of legacy system as defined in 4.3 -4.6

[1] Independent Verification and Validation - role and responsibility are monitoring, reporting, supporting and participating but not performing – applies in all phases and tasks.

6 Capabilities and Best Practices in System Development

OBJECTIVE:

Identify the capabilities needed by the system development team in the development of an ITS project. Describe the Capability Maturity Model used to assess the capabilities of organizations for ITS project development.

Intelligent Transportation Systems development often requires that the project owner put together or select a team to define, design, develop and deploy an ITS project. Candidate teams are evaluated against a set of criteria such as past performance, dedication of key staff to the project, approach to the project, knowledge of the project, and schedule for completion. Most of the time cost is not considered until the evaluation of capabilities has been made and a "short list" of teams has been determined. This has been the traditional approach in the past. In recent years new criteria in the area of capabilities in software and systems development have emerged and now there is the ability to evaluate candidate development teams or internal agency organizations based on their capability to perform software and systems development. These criteria can be found in the tool called Capability Maturity Model Integration (CMMI).

The following is a description of CMMI.

Background:

The following is an excerpt from "CMMI Distilled: A Practical Introduction to Integrated Process Improvement", SEI Series in Software Engineering, Ahern, Clouse, Turner, Addison-Wesley 2001.

> "Model-based process improvement involves the use of a model to guide the improvement of an organization's processes. Process improvement grew out of the quality management work of Deming[2], Crosby[3], and Juran[4] and this work was aimed at increasing the capability of work processes. Essentially, process capability is the inherent ability of a process to produce planned results. As the capability of the process increases, it becomes predictable and measurable, and the most significant causes of poor quality and productivity are controlled or eliminated. By steadily improving its process capability, the organization 'matures."

The early 1990's saw a proliferation of models for process assessment that included: acquisition, people, security, integrated product development, software, systems development, and project framework, in addition to ISO 9000 series, just to name a few. This created a quagmire of process standards and quality models (see http://www.software.org/quagmire). To eliminate inconsistencies and duplication, and provide a common framework, terminology and focus, in 1997 Capability Maturity Model Integration (CMMI) was initiated by the U.S. Department of Defense and the National Defense Industrial Association (NDIA). They teamed with the Software Engineering Institute at Carnegie Mellon to integrate the pertinent models for systems development together into a single model. This is now called CMMI (Capability Maturity Model Integration). The CMMI model used source material from Software (SW-CMM, draft version 2c), Systems Engineering (EIA/IS 731) and integrated product and process development (IPD-CMM, version 0.98). The team that put CMMI together included authors from the source models and other key industry experts. The final version was completed by 2000 and the most current version of CMMI can be downloaded free from http://www.sei.cmu.edu/cmmi/. CMMI is the model superseding previous assessment tools such as SW-CMM and systems engineering EIA 731.

Best Practice Areas:

Figure 6-1 illustrates how CMMI has integrated the best practices from source material into 24 Process Areas (EIA 731 has 19 of these and calls them Focus Areas) or Best Practices. In CMMI these process areas are divided into four categories as illustrated. These process areas cover the "waterfront" of best practices needed for systems development. The pyramid indicates that the organizational processes support the next level up and so on to the engineering processes. It also indicates an organization wide activity and key to successful systems development.

---

[2] Deming, W. Edwards, *Out of the Crisis*, Cambridge, MA; MIT Center for Advanced Engineering, 1986

[3] Crosby, P.B. *Quality is Free*. New York: McGraw-Hill, 1979

[4] Juran, J.M. *Juran on Planning for Quality*, New York; MacMillan, 1988

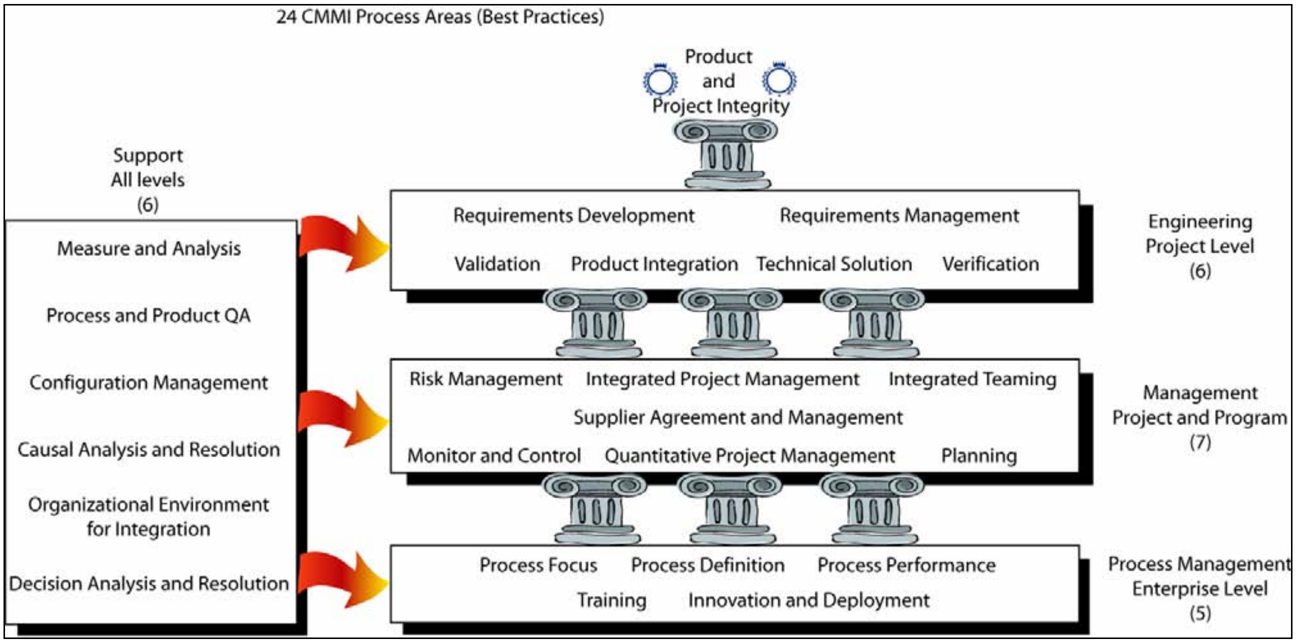


Figure 6-1 CMMI Process Areas and Categories

As illustrated, a set of best practices is associated with each of the following categories: engineering, management, support and process management. For each of the 24 process areas illustrated, there is a set of practices that are required to be performed by the organization to demonstrate a capability and achieve a level of maturity.

Rating Systems:

So how is CMMI used? It is a rating system that was developed by the Software Engineering Institute (SEI) and used by Software and Systems development firms to rate how well their organization performs software and systems development. It is also used by system owners as an evaluation tool for the selection of a candidate systems development organization. This rating is done in 2 different ways – as a staged representation illustrated in Figure 6-2 and continuous representation illustrated in Figure 6-3.

1) Staged representation - provides a single number 0-5 for an organization and is an indicator on how well they perform software or systems development

- *Level 0* (not on scale) - means that no processes are documented or followed
- *Level 1 (Initial)* - Competent people, heroics of the individuals characterize the completion of projects. Processes are known and understood but the performance is sometimes unpredictable, poorly controlled, and reactive
- *Level 2 (Repeatable)* - Basic project management is performed, some configuration, requirements, planning and control is performed. Practices at the project level only and reactive. Characterized as a good project team working together and produces repeatable results from project to project.
- *Level 3 (Defined)* - indicates that the organization has standardized documented processes and follows them. The organization has documented set of processes they are proactive in the execution of the processes.
- *Level 4 (Managed)* - indicates that the organization has statistical methods for analyzing the processes that are performed. The processes are measured and controlled.
- *Level 5 (Optimizing)* - Organizations have continuous process improvement. The organization has a focus on process.

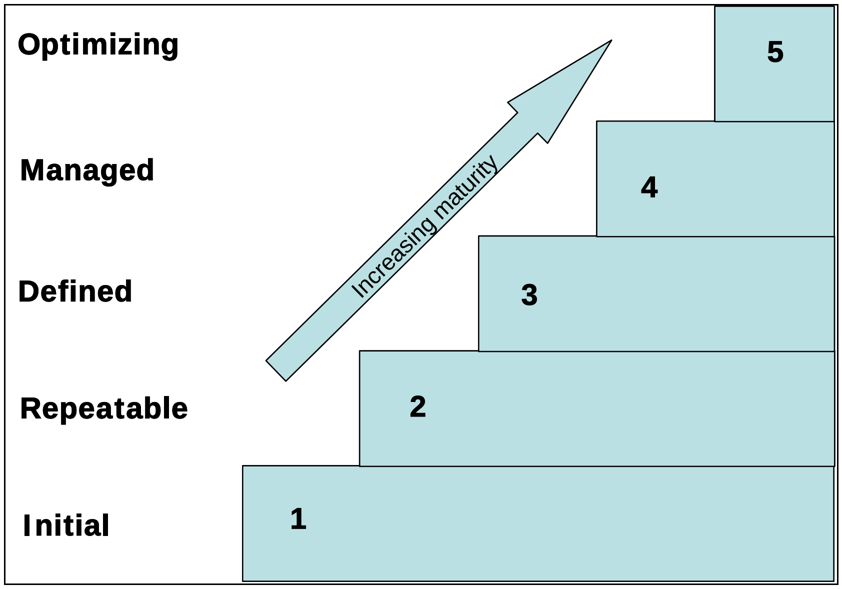


Figure 6-2 Staged View or Representation of CMMI

2) Continuous representation - is used to focus on specific best practices and is not concerned about an overall rating. In this case an organization may focus on 20 or 24 best practices that are critical for the organization. The focus for this example is on performing the 20 best practice areas at a level 3 or higher. The others are not relevant to the organization. Figure 6-3 illustrates a fractional score for configuration management. This is done in the continuous representation because an assessment credits an organization for individual sub-processes. For example, if 8 sub-processes make up configuration management and 4 of them are performed at a level 3 while the other 4 are performed a level 2 then the organization will get credit for a 2.5 rating for the configuration management process. This results in fractional scores for individual processes in the continuous representation.

Levels of Maturity:

CMMI is a single model that has two representations or views, staged and continuous, as discussed above. An organization can choose which representation to use for process improvement and achieve a level of maturity for their organization.

Some organizations would prefer to use the staged representation and another organization the continuous representation depending on their goals and objectives. For example, a company that provides systems development services may elect to use the staged view since the results would be a simple number that identifies the organizations maturity level (1-5) as illustrated in Figure 6-2. Other organizations may elect to use continuous representation to illustrate a "profile" of maturity across the process areas as illustrated in Figure 6-3. These organizations may be more interested in achieving a profile that addresses specific needs. For example, it may be appropriate for a large agency that develops their own systems to use the continuous view in order to achieve maturity in specific areas while other areas may not be applicable to them.

It should be noted that in some cases the higher levels of maturity are not needed or warranted and an organization may elect to stay at a level 2 or 3. The processes involved to achieve the higher levels of maturity (3, 4 and 5) may be too expensive for the return, or the domain of practice does not require it.

The following is an example of how the stages of maturity build on each other. A development company at a level 2 CMMI (staged representation), means that they have a set of repeatable processes (e.g., estimating the cost for developing software). If a company advertises that they are a level 3 (staged representation), that implies that they not only have repeatable processes required for level 2, but also have defined and documented processes required for level 3. In the staged representation, each level of competency builds on the previous level. CMMI also provides a way to map the continuous representation into a staged representation.

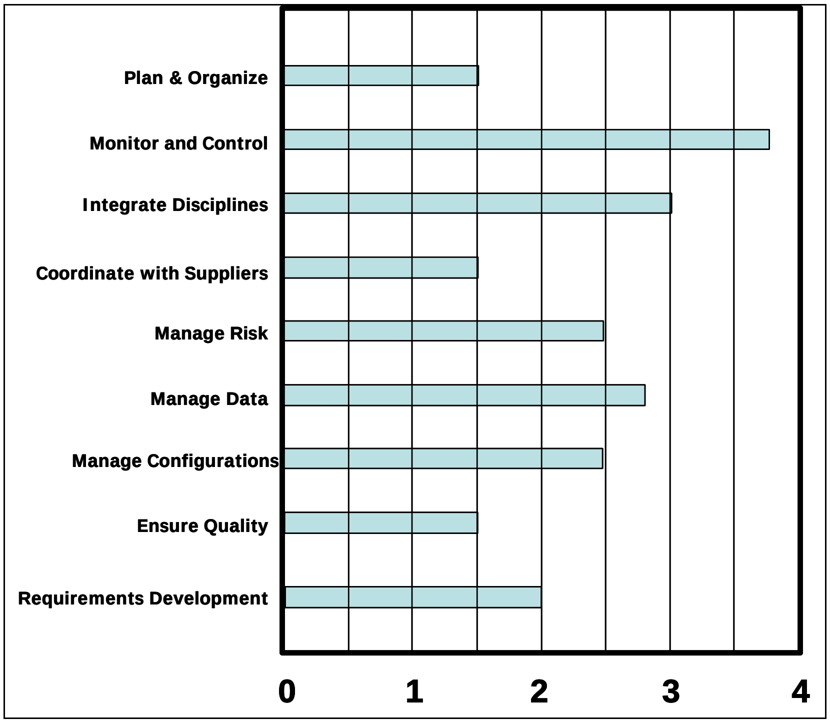


Figure 6-3 Continuous View or Representation of CMMI with Nine Example Process Areas

How would the ITS project owner use CMMI for contracting purposes?

Armed with this knowledge, a project sponsor can develop and publish a Request for Qualifications (RFQ) using a level of maturity for a systems development team as one of the evaluation criterion. When verified, this would demonstrate the development team's capabilities. The caution here is, for large companies, the maturity level may refer to a specific team within a company and does not apply to the entire company unless the whole company performed on the assessed project at the desired maturity level. So when reviewing the qualifications, make sure that the team proposing at a maturity level is the same team that was assessed.

How would an organization use CMMI to improve internal processes?

Within CMMI there is a process set up for doing CMMI assessments; this is called Assessment Requirements for CMMI (ARC). There are trained assessors who will evaluate an organization and provide a report. The following are the major requirements for assessments:

- Responsibilities – lists responsibilities of the assessment sponsor and assessment team
- Assessment method defined
- Planning and preparing for the assessment
- Assessment data collection
- Assessment data consolidation and validation
- Rating
- Reporting results

The team usually does an interview of the assessment sponsors to determine the goals and objectives of the assessment and will tailor the assessment for that purpose.

Specific projects are identified and those project teams are interviewed using a set of assessment tools such as questionnaires and interviews. Once the results are validated, a report is generated that identifies the level of maturity and/or areas of needed improvement.

Types of assessments:

The following are the types of assessments that can be done:

- ***Formal SEI -ARC*** allows the following:
  - o Class A – Full comprehensive method
  - o Class B – Less comprehensive partial self- assessment
  - o Class C – Quick look (check specific areas)
- ***Internal Assessments*** are done in-house by usually another department or a quality team, with assessment capabilities. This assessment is carried out in the same way as SEI assessors would but it is not officially

reported to SEI when completed. This type of assessment is less political and likely to be more realistic than a formal assessment that gets advertised outside the organization.

- Internal assessment can be carried out in accordance with the Formal SEI-ARC

- ***Mini assessments*** are usually internal assessment and perform quick look at key area. These very popular assessments are inexpensive to conduct, using internal resources.

The cost of these assessments varies greatly and depends on the type of assessment and tailoring requested, and size of the team. A minimum cost would be approximately $30K-60K (Quick look and mini-assessment) to several hundred thousands of dollars. (Class A Formal full comprehensive method)

Formal and internal assessments are common for Defense, Federal Aviation Administration, Department of Energy, and for firms providing development services for banking and some information technology firms.

For transportation agencies, these types of assessments are not well known or widely used. So the question is, have any ITS development teams been appraised against CMMI or even one of the earlier models for systems engineering (SE-CMM) or software (SW-CMM)? The answer is most likely very few. The reason is that agencies are, for the most part, unaware of these tools and, as such, have not made it one of the evaluation criteria in RFP's or RFQ's. Since there has not been a significant push from the agencies to make this happen, development firms have not offered this in their proposals. Hopefully, as a result of this Guidebook, criteria for CMMI levels of maturity will start showing up in RFQ's, RFP's and RFI's. Over time it should be common for ITS development teams to perform at a minimum level 2 and preferred level 3 on the CMMI staged representation. This will provide the project sponsor the confidence that the selected development team has the capability for performing well on ITS projects, thereby reducing project risks.

TIP

When a system integrator makes the decision to achieve a CMMI maturity level 2 the average time is from 1-2 years to document, train and implement the needed processes, and then the assessment, takes place after the level 2 practices have been applied to a real project. This may take another 2 years (assuming an 18-24 month project). In summary, it may take from 3-5 years for a system integrator starting a level 0 to be assessed and recognized as being at CMMI level 2.

In the short term, it is recommended that evidence of in-progress work toward a CMMI level of maturity be shown as a demonstration of a best effort over the next 3 to 5 years. For example this could be done by providing documented processes and procedures (drafts or final) showing staff with appropriate certifications or the systems integrators process improvement. After this period, System Owners would give this criterion more weight in the qualification evaluation process.

Systems Engineering Technical Assistance (SETA) consultants or systems managers may not be currently considering any formal assessment to be performed. An internal assessment using the continuous representation would be recommended, though. As an alternative, staff can demonstrate their expertise through professional certification programs like the INCOSE Certified Systems Engineering Professional (CSEP) and Project Management Institute (PMI) certification.

The following table identifies the suggested capabilities profile and best practices that each would consider having at a minimum for ITS development

Table 6-1 Suggested Capabilities Table for ITS

Best Practices (CMMI)	Systems Owner (Project Sponsor)	Systems Engineering Assistance: (In-house, Consultant, System Manager)	Development Team (In-house, Systems Integrator)
Engineering			
Requirements Management	1	1	2
Requirements Development	1	2	2
Technical Solution	1	1	2
Product Integration	1	1	2
Verification	1	2	2
Validation	1	2	2
Project Management			
Project Planning	2	2	2
Project Monitoring and Control	2	1	2
Supplier Agreement Management	2	1	2
Integrated Project Management	1	1	2
Risk Management	3	1	3
Integrated Teaming	1	1	2
Quantitative Project Management	1	1	1
Support			
Configuration Management	3	1	3
Process and Product Quality Management	1	1	1
Decision Analysis and Resolution	1	2	2
Organizational Environment for Integration	1	1	2
Causal Analysis and Resolution	1	2	2
Process Management			
Organizational Process Definition	2	1	3
Organizational Process Focus	2	1	2
Organizational Training	2	1	1
Organizational Process Performance	2	1	1
Organizational Innovation and Deployment	2	1	1

Legend for Table

Level 1 Initial	Level 2 Repeatable	Level 3 Defined

7 Appendices

These appendices contain a wide variety of useful information that summarizes and supplements the earlier parts of the Guidebook. The sections on **Glossary**, **Definition of Terms**, and on **References** conveniently summarize for the reader in one place all of these items used in the guidebook. The sections on **Project Applications**, **Contract Guidance**, and **Systems Engineering Documentation Guidance** supplement that material found in the other parts of the Guidebook and provide practical, real-world guidance to the reader.

The following sections may be found in the appendices:

7.1 Glossary and acronyms used in the Guidebook. Like any discipline, ITS has its own language. Communications between practitioners will be much clearer if that language is used consistently.

7.2 References to other documents contained in the Guidebook. These references will give the reader the ability to explore subjects of specific interest in much more detail.

7.3 Project Application, typical projects and the systems engineering activities that are necessary to support them. These three generic projects were chosen to demonstrate the range of application of the systems engineering processes.

7.4 Contract Guidance for issues of special importance to Intelligent Transportation Systems. The two topics covered are preparation of a Request for Proposal and recommendations on Intellectual Property Rights for software and documentation.

7.5 Systems Engineering Documentation Guidance for commonly occurring documents in the systems engineering processes. These sections provide information on the purpose and content of several technical documents that, in one form or another, most ITS projects will need. They also show how these documents are tailored to the specific needs of your project

7.1 Glossary and Acronyms

7.1.1 Glossary

This glossary and acronym list includes all of the key terms and acronyms that are used in this guidebook, as well as others that often appear in systems engineering. While these are many of the definitions that can be used, each project will have its own set of terms that need to be defined and adopted as part of the project formation and initial tasks.

Acceptance - An action by an authorized representative of the acquirer by which the acquirer assumes ownership of products as a partial or complete performance of contract.

Acceptance criteria - The criteria a product must meet to successfully complete a test phase or meet delivery requirements.

Acceptance test - Formal testing conducted to determine whether or not a system satisfies its acceptance criteria and to enable the acquirer to determine whether or not to accept the system.

Acquirer - An organization that procures products for itself or another organization.

Approval - Written notification by an authorized representative of the acquirer that the developer's plans, design, or other aspects of the project appear to be sound and can be used as the basis for further work. Such approval does not shift responsibility from the developer to meet contractual requirements.

Architecture - The organizational structure of a system, identifying its components, their interfaces, and a concept of execution among them.

Assembly - A number of parts or subassemblies, or any combination thereof joined together, to perform a specific function and capable of disassembly.

Audit - An independent examination of a work product/process or set of work products/processes to assess compliance with specifications, standards, contractual agreements, or other criteria.

Authentication - The procedure (essentially approval) used by the approval authority in verifying that specification content is acceptable. Authentication does not imply acceptance or responsibility for the specified item to perform successfully.

Baseline - An approved product at a point in time. Any changes made to this product must go through a formal change process.

Certification - A process, which may be incremental, by which a contractor provides evidence to the acquirer that a product meets contractual or otherwise specified requirements.

Commercial off-the-shelf (COTS) software - Commercially available applications sold by vendors through public catalogue listings, not intended to be customized or enhanced. (Contract-negotiated software developed for a specific application is not COTS software.)

Components - Components are the named "pieces" of design and/or actual entities (subsystems, hardware units, software units) of the system/subsystem. In system/subsystem architectures, components consist of subsystems (or other variations), hardware units, software units, and manual operations.

Computer database - see database.

Computer hardware - Devices capable of accepting and storing computer data, executing a systematic sequence of operations on computer data, or producing control outputs. Such devices can perform substantial interpretation, computation, communication, control, or other logical functions.

Computer program - A combination of computer instructions and data definitions that enable computer hardware to perform computational or control functions.

Computer software - See software.

Concept (project concept) – A high-level conceptual project description, including services provided and the operational structure.

Concept exploration – The process of developing and comparing alternative conceptual approaches to meeting the needs that drive the project.

Concept of Operations - A document that defines the way the system is envisioned to work from multiple stakeholder viewpoints (Users including operators, maintenance, management).

Configuration item (CI)- A product such as a document or a unit of software or hardware that performs a complete function and has been chosen to be placed under change control. That means that any changes that are to be made must go through a change management process. A baseline is a configuration item.

Configuration management - A discipline applying technical and administrative direction and surveillance to: identify and document the functional and physical characteristics of Configuration Items (CI's); audit the CI's to verify conformance to specifications, manage interface control documents and other contract requirements; control changes to CI's and their related documentation; and record and report information needed to manage CI's effectively, including the status of proposed changes and the implementation status of approved changes.

Configuration Management Plan - A plan defining the implementation (including policies and methods) of configuration management on a particular program/project.

Contract - A mutually binding legal relationship obligating a seller to furnish the supplies or services (including construction) and a buyer to accept and pay for them. It includes all types of commitments, in writing, that obligate the buyer to an expenditure of appropriate funds. In addition to bilateral agreements, contracts include, but are not limited to, awards and notices of awards; job orders or task letters issued under purchase orders under which the contract becomes effective by written acceptance or performance; and bilateral modifications.

Contractor - An individual, partnership, company, corporation, association or other service, having a contract with a buyer for the design, development, manufacture, maintenance, modification, or supply of items under the terms of a contract.

Control gates - Formal decision points along the lifecycle that are used by the system owner and stakeholders to determine if the current phase of work has been completed and that the team is ready to move into the next phase of the lifecycle.

Cross-cutting activities - Enabling activities used to support one or more of the lifecycle process steps.

Data - Recorded information, regardless of medium or characteristics, of any nature, including administrative, managerial, financial, and technical.

Data product - Information that is inherently generated as the result of work tasks cited in a Statement of Work (SOW) or in a source document invoked in the contract. Such information is produced as a separate entity (for example, drawing, specification, manual, report, records, and parts list).

Database - A collection of related data stored in one or more computerized files in a manner that can be accessed by users or computer programs via a database management system.

Database management system - An integrated set of computer programs that provide the capabilities needed to establish, modify, make available, and maintain the integrity of a database.

Decomposition – The process of successively breaking down the system into components that can be built or procured. Functional and physical decomposition are the key activities that are used. Functional decomposition is breaking a function down into its smallest parts. For example, the function ramp metering decomposes into a number of sub-functions, e.g. detection, meter rate control, main line metering, ramp queuing, time of day, and communications. Physical decomposition defines the physical elements needed to carry out the function. For example, the ramp metering physical decomposition includes loops or video detection, WWV time, fiber or twisted pair for communications, 2070 or 170 controllers, and host computer.

Design - Those characteristics of a system or components that are selected by the developer in response to the requirements.

Detailed Design Document – The product baseline used to develop the hardware and software components of the system.

Developer - An organization that develops products ("develops" may include new development, modification, reuse, reengineering, maintenance, or any other activity that results in products) for itself or another organization.

Development model - A specific portion of the lifecycle model that relates to the definition, decomposition, development and implementation of a system or a part of a system.

Development strategy - The way the development and deployment of the overall system will be carried out. For example, an evolutionary development strategy means that the system will be developed and deployed piece-by-piece over time. These pieces are complete functional units that will perform independently from other functional pieces. Incremental development is the development of pieces that are done concurrently or nearly concurrently by the same or different development teams.

Elicitation – The process to draw out, to discover and to make known so to gain knowledge and information, often used in defining needs.

Enabling products - Products that enable the end product to be developed, supported and maintained. For example, these products typically are the software compilers, prototypes, development workstations, plans, specifications, requirements and training materials.

End products - Products that perform the desired capability e.g. the hardware, software, communications and databases.

End-item - A deliverable item that is formally accepted by the acquirer in accordance with requirements of a detail specification.

Evaluation - The process of determining whether an item or activity meets specified criteria.

Evolutionary development - Breaking a project down into parts and developing them in serial fashion.

Feasibility assessment – A pre-development activity to evaluate alternative system concepts, select the best one, and verify that it is feasible within all of the project and system constraints.

Firmware - The combination of hardware device and computer instructions and/or computer data that resides as read-only software on the hardware device.

Gap analysis - A technique to assess how far current (legacy) capabilities are from meeting the identified needs, to be used to prioritize development activities. This is based both on how far the current capabilities are from meeting the needs (because of insufficient functionality, capabilities, performance or capacity) and whether the need is met in some places and not others.

Hardware - Articles made of material, such as aircraft, ships, tools, computers, vehicles, fittings, and their components (mechanical, electrical, electronic, hydraulic, pneumatic). Computer software and technical documentation are excluded.

Integrated product team - A team consisting of agency and contractor representatives working together.

Integrity – A system characteristic that means that the system's functional, performance, physical and enabling products are accurately documented by its requirements, design, and support specifications.

Intelligent Transportation Systems – A broad range of diverse technologies which, when applied to our current transportation system, can help improve safety, reduce congestion, enhance mobility, minimize environmental impacts, save energy, and promote economic productivity. ITS technologies are varied and include information processing, communications, control and electronics.

Interface - The functional and physical characteristics required to exist at a common boundary - in development, a relationship among two or more entities (such as software-software, hardware-hardware, hardware-software, hardware-user, or software-user).

Interface control - Interface control comprises the delineation of the procedures and documentation, both administrative and technical, contractually necessary for identification of functional and physical characteristics between two or more configuration items that are provided by different contractors/acquiring agencies, and the resolution of the problems thereto.

Item - A non-specific term used to denote any product, including systems, subsystems, assemblies, subassemblies, units, sets, accessories, computer programs, computer software or parts.

Legacy system - The existing system to which the upgrade or change will be applied.

Lifecycle - The end-to-end process from conception of a system to its retirement or disposal.

Lifecycle model – A representation of the steps involved in the development and other phases of an ITS project.

Market packages - Potential products or subsystems that address specific services (as used in an ITS architecture).

Metrics - Measures used to indicate progress or achievement.

Model - An abstraction of reality. Examples: A road map is an abstraction of the real road network. A globe is a model of the world. A simulation is a dynamic model of a time sequence of events.

Module - A self-contained part of a hardware item designed as a single replaceable unit, with a specific integral electronic function. It should require no installation other than mechanical mounting and completion of electrical connection.

National ITS Architecture - A general framework for planning, defining, and integrating ITS. It was developed to support ITS implementations over a 20-year time period in urban, interurban, and rural environments across the country. The National ITS Architecture is available as a resource for any region and is maintained by the USDOT independently of any specific system design or region in the nation.

Needs assessment - An activity done early in system development to ensure that the system will meet the most important needs of the project's stakeholders, specifically that the needs are well understood, deconflicted, and prioritized.

Non-conformance - The failure of a unit or product to conform to specified requirements.

Operational baseline - The system that is currently in use, including all of the design, development, test, support and requirements documentation.

Operational concept - The roles and responsibilities of the primary stakeholders and the systems they operate.

Part - One piece, or two or more pieces joined together which are not normally subjected to disassembly without destruction or impairment of designed use (examples: gear, screws, transistors, capacitors, integrated circuits.

Performance - A quantitative measure characterizing a physical or functional attribute relating to the execution of a mission/operation or function.

Policy - A guiding principle, typically established by senior management, which is adopted by an organization or project to influence and determine decisions.

Process - An organized set of activities

Product - A product is a given set of items. The set could consist of system, subsystem, hardware, or software items and their documentation.

Project - An undertaking requiring concerted effort, which is focused on developing and/or maintaining a specific product. The product may include hardware, software, and other components. Typically a project has its own funding cost accounting and delivery schedule with the acquirer (customer).

Project architecture – High-level design

Project lifecycle - see Lifecycle

Project Plan – A description (what is to be done, what funds are available, when it will be done and by whom) of the entire set of tasks that the project requires.

Qualification testing - Testing performed to demonstrate to the acquirer that an item, system or subsystem meets its specified requirements.

Quality assurance - A planned and systematic pattern of all actions necessary to provide adequate confidence that management and technical planning and controls are adequate to establish correct technical requirements for design and manufacturing, and to manage and design activity standards, drawings, specifications, or other documents referenced on drawings, lists, or technical documents.

Reengineering - The process of examining and altering an existing system to reconstitute it in a new form. This may include reverse engineering (analyzing a system and producing a representation at a higher level of abstraction, such as design from code), restructuring (transforming a system from one representation to another at the same level of abstraction), recommendation (analyzing a system and producing user and support documentation), forward engineering (using software products derived from an existing system, together with new requirements, to produce a new system), and translation (transforming source code from one language to another or from one version of a language to another).

Regional ITS Architecture - A specific regional framework for ensuring institutional agreement and technical integration for the implementation of ITS projects in a particular region.

Requirements - The total consideration as to WHAT is to be done (functional), HOW well it is to perform (performance), and under WHAT conditions (environmental and non-functional).

Reverse engineering - The process of documenting an existing Intelligent Transportation Systems functional (what it does – requirements), physical (how it does it – design) and support (the way it was built and maintained – enabling products) characteristics.

Risk management - An organized process to identify what can go wrong, to quantify and access associated risks, and to implement/control the appropriate approach for preventing or handling each risk identified.

Software - Computer programs and computer databases. Note: Although some definitions of software include documentation, it is now limited to the definition of computer programs and computer databases.

Software development - A set of activities that results in software products. Software development may include new development, modification, reuse, re-engineering, maintenance, or any other activities that result in software products.

Specification - A document that describes the essential technical requirements for items, materials or services including the procedures for determining whether or not the requirements have been met.

Stakeholders - The people for whom the system is being built, as well as anyone who will manage, develop, operate, maintain, use, benefit from, or otherwise be affected by the system.

Statement of Work (SOW) - A document primarily for use in procurement, which specifies the work requirements for a project or program. It is used in conjunction with specifications and standards as a basis for a contract. The SOW will be used to determine whether the contractor meets stated performance requirements.

Subcontractor - an individual, partnership, corporation, or an association that contracts with an organization (i.e., the prime contractor) to design, develop, and/or manufacture one or more products.

Suppliers - the term 'suppliers' includes contractors, sub-contractors, vendors, developers, sellers or any other term used to identify the source from which products or services are obtained.

Synthesis - The translation of input requirements (including performance, function, and interface) into possible solutions (resources and techniques) satisfying those inputs. This defines a physical architecture of people, product and process solutions for logical groupings of requirements (performance, functions, and interface) and their designs for those solutions.

System elements - A system element is a balanced solution to a functional requirement or a set of functional requirements and must satisfy the performance requirements of the associated item. A system element is part of the system (hardware, software, facilities, personnel, data, material, services, and techniques) that, individually or in combination, satisfies a function (task) the system must perform.

System - An integrated composite of people, products and processes, which provide a capability to satisfy a stated need or objective.

Systems engineering - An inter-disciplinary approach and a means to enable the realization of successful systems. Systems engineering requires a broad knowledge, a mindset that keeps the big picture in mind, a facilitator and a skilled conductor of a team.

System specification - A system level requirements specification. A system specification may be a system/subsystem specification, Prime Item Development Specification, or a Critical Item Development Specification.

Tailoring – Planning systems engineering activities that are appropriate and cost-effective for the size and complexity of the project. It may be based on cost, size, the number of stakeholders, the supporting relationships between them, complexity of systems (large number of interfaces to other systems, a large number of functions to perform, or the degree of coupling between systems.), level of ownership of system products (custom development of software owned by the agency or commercial off the shelf products), existing software products, resources, risks.

Technical (formal) reviews - A series of system engineering activities by which the technical progress on a project is assessed relative to its technical or contractual requirements. The formal reviews are conducted at logical transition points in the development effort to identify and correct problems resulting from the work completed thus far before the problem can disrupt or delay the technical progress. The reviews provide a method for the contractor and procuring activity to determine that the identification and development of a CI have met contract requirements.

Testable - A requirement or set of requirements is considered to be testable if an objective and feasible test can be designed to determine whether each requirement has been met.

Trade-off Study - An objective evaluation of alternative requirements, architectures, design approaches, or solutions using identical ground rules and criteria.

User - The organization(s) or persons within those organization(s) who will operate and/or use the system for its intended purpose.

User services - A catalog of features that could be provided by an ITS project (as used in an ITS architecture).

Validation - The process of determining that the requirements are the correct requirements and that they are complete. The system lifecycle process may use requirements that are derived requirements in system validation.

Vendor - A manufacturer or supplier of an item.

Verification - The process of determining whether or not the products of a given phase of the system/software lifecycle fulfill the requirements established during the preceding phase.

Work breakdown structure - (WBS) A product-oriented listing, in family tree order, of the hardware, software, services and other work tasks, which completely defines a product or program. The listing results from project engineering during the development and production of a materiel item. A WBS relates the elements of work to be accomplished to each other and to the end product.

7.1.2 Acronyms

AAA American Automobile Association
AASHTO American Association of State Highway and Transportation Officials
AIAA American Institute of Aeronautics and Astronautics
ANSI American National Standards Institute
ASTM American Society for Testing and Materials
ATIS Advanced Transportation Information System
ATMS Advanced Transportation Management System
C2C Center to Center
C2F Center To Field
CAIV Cost As an Independent Variable
CCTV Closed-Circuit Television
CDR Critical Design Review
CE Concept Exploration
CEA Consumer Electronics Association
CFR Code of Federal Regulations
CG Control Gate
CI Configuration Item
CM Configuration Management
CMM Capability Maturity Model
CMMI Capabilities Maturity Model Integrated
CMS Changeable Message Sign
COCOMO Constructive Cost Model
ConOps Concept of Operations
CORBA Common Object Request Broker Architecture
COTS Commercial Off-The-Shelf
CR Change Request
DATEX Data Exchange
DBMS Database Management System
DDR Detail Design Review
DOT Department of Transportation
ECP Engineering Change Proposal
EDI Electronic Data Interchange
EIA Electronic Industries Association
FAR Federal Acquisition Regulation
FCA Functional Configuration Audit
FHWA Federal Highway Administration
FPA Function Point Analysis
FSR Feasibility Study Report
FTP File Transfer Protocol
GUI Graphical User Interface
HLD High-Level Design
ICD Interface Control Documentation
ICWG Interface Control Working Group

IEEE Institute of Electrical and Electronics Engineers
IKIWISI I Know It When I See It
INCOSE International Council of Systems Engineering
IPT Integrated Product Teams
ISO International Organization for Standardization
IT Information Technology
ITE Institute of Transportation Engineers
ITIP Interregional Transportation Improvement Plan
ITS Intelligent Transportation System(s)
IV&V Independent Verification and Validation
KPP Key Performance Parameter
MOE Measure of Effectiveness
MOP Measure of Performance
MPO Metropolitan Planning Organization
NA Not Applicable
NASA National Aeronautics and Space Administration
NCOSE National Council on Systems Engineering (now INCOSE)
NDI Non-Developmental Item
NEMA National Electrical Manufacturers Association
NIST National Institute of Standards and Technology
NTCIP National Transportation Communications for ITS Protocol
O&M Operations and Maintenance
ORB Object Request Brokers (programming middleware)
PD Product Development
PDR Preliminary Design Review
PM Program Manager
PPP Point-to-Point Protocol
QFD Quality Function Deployment
RFP Request for Proposal
RFQ Request for Quotation
RTIP Regional Transportation Improvement Plan
ROW Right Of Way
SAE Society of Automotive Engineers
SDR System Design Review
SE Systems Engineering
SEI Software Engineering Institute (Carnegie Mellon University)
SEMP Systems Engineering Management Plan
SERF Systems Engineering Review Form
SETA Systems Engineering Technical Assistance
SI Software Item
SOW Statement of Work
SRR System Requirements Review
STIP Statewide Transportation Improvement Plan
SW Software
T & E Test & Evaluation

TIP	Transportation Improvement Plan
TMC	Traffic Management Center
TR	Technical review
TRR	Test Readiness Review
UML	Unified Modeling Language
WAN	Wide Area Network
WBS	Work Breakdown Structure
XML	Extended Mark-up Language

7.2 Text, Papers and Website References

7.2.1 Systems Engineering References

The following is a compilation of systems engineering, requirements and ITS references that will be helpful to the reader. This is organized with general systems engineering texts, followed by requirements engineering texts, then studies and papers, and finally ITS references.

Title: Introduction to Systems Engineering
Author: Andrew P. Sage, James E. Armstrong , Jr.
Copyright: 2000
Publisher: Wiley
ISBN: 0-471-02766-9
Comment: This is a good introductory college level textbook. It has problem sets at the end of each chapter. There are also many bibliographic references for each chapter.

Title: Systems Engineering
Author: Andrew P. Sage
Copyright: 1992
Publisher: Wiley
ISBN: 0-471-53639-3
Comment: This is a good introductory college level textbook. It has problem sets at the end of each chapter. There are also many bibliographic references for each chapter.

Title: The Engineering and Design of Systems: Models and Methods
Author: Dennis M. Buede
Copyright: 2000
Publisher: Wiley
ISBN: 0-471-28225-1
Comment: This is a good advanced college level textbook. It has problem sets at the end of each chapter. There are also many bibliographic references for each chapter.

Title: Handbook of Systems Engineering and Management
Editors: Andrew P. Sage, William B. Rouse
Copyright: 1999
Publisher: Wiley
ISBN: 0-471-15405-9
Comment: This is a compendium of works from 40 contributing authors. At the end of each chapter are lists of additional references and a bibliography supporting the work.

NOTE: These books are a part of the publisher's series of books on systems engineering. At this time there are seventeen works in this series. The principal authors are from George Mason University Systems Engineering & Operations Research Department. This school offers B.S. and M.S. degrees in this field.

Title: Systems Engineering Guidebook: A Process for Developing Systems and Products
Editors: James N. Martin, A. Terry Bahill
Copyright: 1999
Publisher: Wiley
ISBN: 0849378370
Comment: This is a compendium of works from many contributing authors.

Title: Systems Engineering and Analysis
Author: Benjamin S. Blanchard, Wolter J. Fabrycky
Copyright: 1998
Publisher: Prentice Hall
ISBN: 0131350471

Title: Systems Engineering Management
Author: Benjamin S. Blanchard
Copyright: 1997
Publisher: Wiley
ISBN: 0471190861

Title: Systems Engineering: Coping with Complexity
Authors: Jackson, Brook, Stevens, Arnold
Copyright: 1998
Publisher: Prentice Hall
ISBN 0130950858
Comment: Excellent reference. Integration of Systems Engineering activities, e.g. CM, Requirement Engineering, Verification and Validation.

Title: Visualizing Project Management
Author: Forsberg, Mooz, Cotterman
Copyright: 2000
Publisher: Wiley
ISBN: 047135760
Comment: Excellent reference on the Vee Development Model and the integration of Project Management and Systems Engineering. The CD that comes with it is very good.

Title: CMMI Distilled: A practical Introduction to Integrated Process Improvement.
Author: Ahern, Clouse, Turner
Copyright: 2001
Publisher: Addison-Wesley
ISBN: 0201735008
Comment: The CMMI is the replacement for SECAM and integrates Software Engineering, Systems Engineering and Integrated Product Team. Good introduction on CMMI Capabilities Maturity Model.

Title: Systems Engineering Guidebook
Author: James N. Martin
Copyright: 1997
Publisher: CRC Press
ISBN: 0849378370
Comment: Jim leads the standards activity for INCOSE. This book has good information that amplifies EIA 632.

Title: Systems Thinking, Systems Practice
Author: Peter Checkland
Copyright: Reprinted November 2000
Publisher: Wiley
ISBN: 0471986062
Comment: Considered a classic work in Systems Engineering and Systems Thinking.

Title: Systems Engineering Principles and Practice
Author: Alexander Kossiakoff and William N. Sweet
Copyright: 2003
Publisher: Wiley
ISBN: 0471234435
Comment: Good general Systems Engineering overview.

Title: Management of Systems Engineering
Author: Wilton P. Chase
Copyright: 1984
Publisher: Krieger Publishing Co.
ISBN:0 89874-682-5
Comment: Basic Systems Engineering Management practices

7.2.2 Requirements Engineering References

Title: Software Requirements Styles and Techniques
Author: Soren Lauesen
Copyright: 2002
Publisher: Addison Wesley
ISBN:0 201 74570 4
Comment: Good elicitation techniques

Title: Requirements Engineering
Author: R.J. Wieringa
Copyright: 1996
Publisher: Wiley
ISBN:0 471 95884 0
Comment: Describes a number of requirements analysis techniques

Title: System Requirements Engineering
Author: Loucopoulos and Karakostas
Copyright: 1995
Publisher: McGraw-Hill
ISBN:0 07 707843 8
Comment: Discussion on Modeling Requirements

Title: Requirements Engineering: A Good Practice Guide
Author: Sommerville and Sawyer
Copyright: 1997
Publisher: Wiley
ISBN:0 471 97444 7
Comment: Good layout, easy to follow

Title: Requirements Engineering Process and Techniques
Author: Kotonya and Sommerville
Copyright: 1998
Publisher: Wiley
ISBN:0 471 97208 8
Comment: Update of the Requirements Engineering, with Object Technology added

Title: INCOSE Systems Engineering Handbook Version 2a
Author: INCOSE
Copyright: 2004
Publisher: INCOSE.
ISBN:
Comment: Compilation of works for the technical processes for systems engineering.

Reference Standards and Papers for Systems and Software Engineering

Title: INCOSE Symposium papers
Author: Various
Copyright: INCOSE
Publisher: INCOSE (Available via http://www.incose.org
ISBN:
Comment: 10 years of papers – From the initial INCOSE conference, a wealth of SE information, available on CD.

Chaos report – The Standish report - http://www.standishgroup.com/sample_research/chaos_1994_1.php

Twenty-five NASA program profiles – NASA HQ – Visualizing Project Management second edition Forsberg, Mooz and Cotterman 2000 pg. 82.

CMMI Systems, Software and Integrated Product Team Capability Maturity Model. http://www.sei.cmu.edu/cmmi

Software Acquisition CMM http://www.sei.cmu.edu/programs/acquisition-support/

FAA's CMM http://www.faa.gov/aio/common/documents/iCMM/ref/usingiCMM.htm

Standards can be purchased at (http://global.ihs.com/)

EIA/IS 632, Draft Standard: Processes for Engineering a System

ISO/IEC 15288 System Lifecycle Processes

ISO 10007 Quality management – Guidelines for Configuration Management 1995

IEEE 1220-1994, IEEE Trial-Use Standard for Application and Management of the Systems Engineering Process.

EIA/IS 731 (SE-CMM): Bate, Roger, et. al., Systems Engineering Capability Maturity Model

IEEE 1362-1998, IEEE Guide for Information – System Definition – Concept of Operations Document.

IEEE 830-1993, IEEE Recommended Practice for Software Requirements Specifications

IEEE 1012-1986, IEEE Standard for Software Verification and Validation Plans

IEEE 1233 - IEEE Guide for Developing System Requirements Specifications

Institute of Electrical and Electronics Engineers, IEEE 1362 IEEE Guide for Information Technology – System Definition – Concept of Operations Document

American National Standards Institute / American Institute of Aeronautics and Astronautics, ANSI / AIAA G-043-1992 Guide for the Preparation of Operational Concepts Documents

IEEE STD 1233 IEEE Guide for Developing System Requirements Specifications

Electronic Industries Alliance, EIA 649 National Consensus Standard for Configuration Management

IEEE 1028-1988 Standards for Review and Audit

IEEE 1471-2000 Recommended Practice for Architectural Descriptions of Software-Intensive Systems

IEEE 1012-1998 Software Independent Verification and Validation

Unified Modeling Language, information available at http://www.omg.org

Integrated Method for Information Modeling (IDEF), information available at http://www.idef.com

References for Intelligent Transportation Systems and Transportation policies and procedures:

Federal Final Rule (23 CFR 940 part 11) http://www.access.gpo.gov/nara/cfr/waisidx_03/23cfr940_03.html

National ITS Architecture - http://www.its.dot.gov/arch/arch.htm.

State of California's Statewide Information Management Manual (SIMM)

State of California's State Administrative Manual (SAM) http://sam.dgs.ca.gov/default.htm

California's SAM 4819.35 (dated 6/03)

Department of General Services of the State of California - http://sam.dgs.ca.gov

SEMP guidelines prepared by the Caltrans Office of Local Assistance, see the web page at: http://www.dot.ca.gov/hq/LocalPrograms/lam/prog_g/g12othr.pdf

The NTCIP Guide updated version 3 http://www.ntcip.org

7.3 Project Examples

The following tables describe the degree of systems engineering that would be applied to three example ITS projects. These tables also provide a guide on the level of effort for each phase of the project. It should be noted that these are only estimates and that each project, even if they are similar to the ones listed, will need to be evaluated on its own merits. The following is a brief description of each of the projects that are listed in the tables to follow:

Project Example 1 – Adding field elements to an existing system – this example adds changeable message signs to an existing system. The point of this example is to show that cost is not necessarily a driver in the amount of systems engineering needed. A 10 million dollar project may need less systems engineering than a $500K project. Also, this example applies to field cameras, ramp metering, intersection controllers, or detection.

Project Example 2 – Adding new functionality to an existing system – this example builds on example 1 **-** the changeable message signs, we add another requirement of sharing control of the signs with a partnering agency. In this example, the existing control software was not designed for this requirement and injected typical institutional issues that ITS projects face in developing regional systems. The point of this example was that the requirement for sharing adds significant risk the project. Even though the estimated cost of the software is small compared to the cost of the changeable message signs, the project risk is driven by the upgrade to the controlling software and the institutional issues. This example also applies to the sharing of field devices such as cameras, signal systems, or the integration of bus priority with signal systems.

Project Example 3 – Implementing a new central management system – this example Upgrades a signal system. This is a typical project and provides a good example of the nominal amount of systems engineering required for a commercial off-the-shelf (COTS) product.

These are just typical activities and estimates of effort. This should not be taken as a “script” to follow. These same projects in your environment may require more or less systems engineering effort.

7.3.1 Example Project 1 - Adding Field Elements to an Existing System

(Please note that the solution given here is for this example only. Other viable solutions may be possible and each must be evaluated for a given project)

A $10 million project to add 30 full matrix changeable message signs (assuming $330,000 per sign) to an existing system that had five of these same signs already deployed. No changes are needed to the existing central or field equipment. The system was initially designed to accommodate these additional signs so no additional software is needed. Assumptions are: 1) The communications and power for the signs have already been deployed under a previous construction project, 2) The initial system has been completed and the system is working, 3) The contractor will deploy the signs, poles and foundations, controllers, and wire the controllers into the signs, 4) The agency will add configuration information about the signs at the central computer, and 5) The construction costs have been included in the cost of the signs.

In this example, even though this is a high dollar amount, very little systems engineering is needed ***because the risks are low and no decisions or trade studies are required***. This same example can be applied to many current ITS projects such as, adding field masters and traffic signals to a traffic signal control system, cameras to an existing surveillance system, or detectors to an existing detection system. Adding elements to existing systems which do not require additional design, coding, or development (other than the construction design needed for the signs and controllers at each location) would require the minimum amount of formal systems engineering. However, it is recommended that updates to existing plans and reviews be performed to ensure that the original design and implementation is not adversely affected as a result of adding the elements.

Process Step	**Estimated Level of Effort**	**Check list of supporting activities**	**Check list of issues**	**Check list of risks**	**Examples**
Feasibility Study	None				Completed and approved as part of the original project.
Planning	Low	☑ Risk mgt ☑ Config mgt	☑ Ensure that the plan(s) are up to date and still applicable.	☑ Changes in staff, stakeholders or institutions, construction, or vendor that may have occurred between the time of the original development and the deployment of these elements. ☑ Vendor defects	Update of the Deployment Plan, and Integration Plans. Construction risks were low, no changes to the designs needed, and the system can be configured to accommodate the additional signs. Vendor has good internal processes, and sign is his standard product.
Development of a Concept of Operations and Validation Plan	None				Reuse of the Validation Plan
Development of System Level Requirements and Verification Plans	None				Reuse of the Verification Plan
Development of High Level Design/Sub-system Requirements and Verification Plans	None				Reuse of the Verification Plan

Process Step	Estimated Level of Effort	Check list of supporting activities	Check list of issues	Check list of risks	Examples
Development of Component Level Design	None				Commercial off-the-shelf product
Hardware/Software Development	None – Low	☑ Technical Review	☑ That the host configuration software is operational and can accommodate the additional signs	☑ Software was not checked out in the original implementation for additional signs	Commercial off-the shelf product Original design and implementation included the additional signs
Unit verification	None				Vendor performed
Unit Integration	None				Vendor performed
Sub-system verification	Low	☑ Technical Review	☑ Verify that controller, signs and communications are working	☑ Defective signs, controller, communications or interface.	Signs and interfaces were checked out and verified at the factory, review of verification data
Sub-system Integration	Low	☑ Technical Review	☑ Coordination of integration activities, integration of controller with communications ☑ Integration of signs and controller	☑ Controller is integrated and working with the communications ☑ Integration of signs and controllers	Use of the same interfaces that were used before. Integration issues will only occur if defects occur in manufacturing of the signs.
System verification	Medium	☑ Technical Review	☑ Verify the host software is configured properly and all functionality is verified on all signs. (Regression) testing on the initial signs may be needed	☑ The added signs or exercising the host software uncovered a defect that was not known at time of initial integration and verification	Reuse of original acceptance Verification Plans – 30 signs to verify
Deployment	Medium	☑ Technical Review	☑ How the signs will be deployed ☑ The resources needed ☑ Normal construction issues	☑ Deployment in a timely manner ☑ Lack of resources to deploy the 30 signs.	Per the Deployment Plan
Validation	None				Validation on original project
Operations and Maintenance	Low	☑ Configuration management	☑ Synchronize the new system configuration with any updates to software, patches, user manuals and fixes with documentation	☑ Loss of the alignment of the documentation with the physical configuration of the system will provide a loss in system integrity.	Update users manuals, as-builds, software documentation if needed.

7.3.2 Example Project 2 - Adding New Functionality to an Existing System

(Please note that the solution given here is for this example only. Other viable solutions may be possible and each must be evaluated for a given project)

Building on Example 1, a new requirement was added. This requirement is that the changeable message signs were to have shared control with a partner agency (Agency B). Primary agency A owns and operates the signs and the host system that controls them. This new requirement was driven by the development of a regional architecture and the existing CMS host system was deployed prior to the regional architecture. The requirement reads: "The changeable message sign system shall share control with agency B". **For this example**, the smaller agency B manages events at two centers. As part of the installation, the primary agency will be installing six (6) signs that would assist agency B for their event management. Agency B would use the CMS in coordination with their local control of traffic signals to divert traffic, appropriately getting the attendees in and out of the event faster and more safely.

New software **may** need to be developed and integrated into the existing system. The project had an initial cost estimated at $10.5 million for the signs, new software, workstations, and communications for the participating agencies and systems engineering activities. With this new requirement, new risks and complexity are introduced relative to example 1. It is recommended that for this example the following systems engineering processes ***(In this example some of the steps needed for Application Category 1 may be incorporated e.g. Technical Reviews, others e.g. Unit Verification by the vendor still needs to be performed***) be used to clearly define and develop the shared control of the CMS.

Process Step	Estimated Level of Effort	Check list of supporting activities	Check list of issues	Check list of risks	Examples
Feasibility	Medium 2-5 pages	☑ Procurement ☑ Trade study ☑ Stakeholder involvement ☑ Elicitation ☑ Risk mgt	☑ Procure the services of systems engineering services ☑ ☑ Documentation and analysis of: ☑ Need to be addressed ☑ Definition of the problem ☑ Scope of the problem ☑ Possible solution concepts ☑ Estimated cost and benefit ☑ Identification of the portion of the regional architecture that this will fulfill. ☑ Institutional issues	☑ Picking a point solution without considering the business case or cost benefit of alternatives. ☑ Selecting a solution that is not appropriate among participating agencies ☑ Proposing a solution that is too costly ☑ Incomplete solutions	**Definition of the problem and need:** "Sharing of CMS by Agency B for event management and to provide alternate routing at the beginning and ending of the event". **Scope:** Agency B needs shared control of 6 CMS that are in the event areas **Feasibility:** Can the existing software be modified to include this new requirement? How much reverse engineering is needed to integrate the new requirement into the existing system? **Trade study and cost benefit:** Evaluate stand-alone systems controlling the signs or integrate software functionality into legacy system at the primary agency. **Institutional issues:** Equipment standards different between the agencies. Limited support staff and maintenance at agency B. **Cost Estimate:** Reverse engineering effort increased the cost of the project to $10.7 million from the original $10.5 million.

Process Step	Estimated Level of Effort	Check list of supporting activities	Check list of issues	Check list of risks	Examples
			☑ Feasibility with existing system(s) ☑ Feasibility with partner agencies ☑ Identify project risks ☑ Technical metrics and performance		**Identified Risks:** Interagency MOUs cannot be signed or delayed Reverse engineering will be more costly than expected Standards and license agreements Security Maintenance Limited solution (not general enough for region)
Planning	Low* to Medium SEMP framework developed 2-3 pages	☑ Stakeholder involvement ☑ Risk mgt ☑ Config mgt ☑ Project planning ☑ Technical reviews	☑ Identification of expected plans for the project ☑ Expected content and quality of the plans ☑ Expectations on the effort needed for the development ☑ Schedule and budget ☑ Monitoring and controlling of effort	☑ Loss of control of the project deliverable, schedule and budget ☑ Missing critical activities ☑ Lack of long term maintenance and operations ☑ Not meeting expectations	The identified technical plans include: Development Plan (Software, Hardware) Integration Plan Deployment Plan Verification and Validation Plans Development team CM Plan Project Plan Operations and Maintenance Plan Configuration Management Plan Risk Management Plan *** Note:** This effort will be low if plan frameworks have already been done, medium effort if they need to be developed.
Concept of Operations and Validation Plan	Medium 5-10 pages	☑ Elicitation ☑ Stakeholder involvement ☑ Risk mgt ☑ Trade studies ☑ Technical reviews	☑ Definition of the way the system will operate and be maintained ☑ Identification of the project level stakeholders e.g. ☑ Maintenance ☑ Supervisors ☑ Operators	☑ Lack of understanding on: ☑ How shared control will operate with limitations ☑ How the system will be maintained ☑ Scope of the project	**How shared control will operate with limitations:** Agency B staff need to monitor the status of the 6 CMS and post messages on alternate routes for local events and emergency traffic conditions Be able to remotely control the signs from the supervisor's home. Limited to the use of pre-developed canned messages by agency B **How the system will be Maintained:** Agency B maintenance is limited and lacks the skills to maintain the communications link. The standard for Agency B is Windows-based workstation or

Process Step	Estimated Level of Effort	Check list of supporting activities	Check list of issues	Check list of risks	Examples
			☑ IT department ☑ Agencies' risk managers ☑ Validation of the system ☑ Limitations of shared control ☑ Alternative operational concepts ☑ Definition of user needs at the project level ☑ Identification of risks ☑ Target performance of the shared control ☑ Revised cost estimate ☑ Agency's normal Operations and Maintenance Standards	☑ Who will be impacted by the control ☑ How the system will be validated ☑ What is needed for shared control ☑ Project risks ☑ Project needs ☑ Operational and Maintenance Standards and agency limitations ☑ City's policies and risks regarding control of the CMS ☑ Missing alternative operational strategies	PC. Staff can install software if installation instructions are provided or there is a standard installation wizard. The primary agency will maintain the host and communications system and provide installation support to agency B. **Operational Standards and Norms** Agency B is a 5-day operation that will support on weekends for events and emergencies from the supervisor's home. The primary agency is a 7-day 24-hour operation. On weekends, if agency B cannot be reached, the primary agency will have the authority to post messages on behalf of agency B (MOUs) as agreed. If any system fault occurs, the primary agency would need to identify and resolve the problem. **Additional risks identified:** Security of the remote link into the system; a security plan will be needed. (Update SEMP with a framework of a Security Plan.) **Validation of the shared control** The transfer of control between agencies will be in accordance with the scenario developed in the conops.
Development of System Level Requirements and Verification Plans	Medium 5-7 pages Requirements and 5-7 page Verification Plan	☑ Stakeholder involvement ☑ Elicitation ☑ Technical reviews ☑ Trade studies ☑ Risk mgt ☑ Config mgt ☑ Traceability	☑ Definition of what the system is to do to support the identified needs ☑ What will be used as the basis for accepting the completed system? ☑ New risks that may be uncovered	☑ Not having a basis to accept the system when completed ☑ Not completely defining what the system is to do ☑ Scope creep ☑ Expectations not met	**Definition of what the system is to do to support the identified needs.** "The changeable message sign system shall share control with agency B" "Agency B shall have remote access to the CMS system" "Remote access to the CMS host shall be secure" "Remote access shall be limited to pre-defined set of messages". **What will be used for the basis of verification and acceptance of the system?** Verification Plan would contain:

Process Step	Estimated Level of Effort	Check list of supporting activities	Check list of issues	Check list of risks	Examples
			☑ New requirements may be needed ☑ Are all the needs addressed? ☑ Is each need addressed completely? ☑ What will be needed to support the development, operations and maintenance? ☑ Project cost ☑ The important things are implemented ☑ Establish a baseline of Requirements that will be used to build the system ☑ Validation of Requirements	☑ Project cost ☑ Losing control and visibility of the development ☑ Requirements not validated by stakeholders	Demonstrate that only a pre-defined set of messages can be displayed. Analysis that the system is secure. **What will be needed to support the development, operations and maintenance?** Users and maintenance documentation shall be provided Installation documentation shall be developed for the host and remote users. **Project Costs** With the additional support documentation and security aspects the project costs have been revised to 10.8 million If only the high priority requirements are implemented, the estimated cost is 10.6 million. **Establish a baseline of requirements that will be used to build the system** Requirements walk-through and a review with the stakeholders is performed for acceptance of the requirements document to establish a system baseline
High Level Design Sub-system Requirements and Verification Plans	Medium 3-5 pages for each of sub-systems	☑ Stakeholder involvement ☑ Trade studies ☑ Risk mgt ☑ Config mgt ☑ Technical reviews ☑ Procurement	☑ What Project Architectures will be viable ☑ Sub-system Requirements ☑ Identification of candidate commercial off the shelf products ☑ Establish a sub-system baseline for each sub-system -	☑ Not deployable ☑ Not maintainable ☑ Inflexible ☑ Agency B staff cannot access internet from home or remotely ☑ Lack of Standards ☑ That the sub-	**What project architectures are viable:** Centralized control with direct dial-in remote links Centralized control with access via internet Centralized control call in/email via operator (man in-the-loop) Distributed workstations direct to field controllers **Recommended Architecture** Centralized control with access via internet (Rational) Remote workstations are platform independent Flexible in a multi-agency environment Maintenance for remotes are minimized VPN technology offers fairly good security

Process Step	Estimated Level of Effort	Check list of supporting activities	Check list of issues	Check list of risks	Examples
			performance ☑ Establish interface standards ☑ Sub-system verification ☑ Add content to Plans as appropriate ☑ Selection of a systems integrator ☑ Project costs and risks	systems are not verified ☑ Project costs ☑ Lack of facilities and services for new functionality ☑ Decomposed incorrectly	Remote software maintained at host (thin client) **Project Costs** Revised cost estimate based on responses from system integrator proposals
Development of Component Level Design	Defined by SEMP – Development Plan	☑ Stakeholder involvement ☑ Trade studies ☑ Risk mgt ☑ Config mgt ☑ Technical reviews ☑ Procurement	☑ Provide content to SEMP ☑ Reverse engineer the legacy system software and restructure for remote application (As-builds) ☑ Recommendations from the system integrator on alternatives ☑ Definition on how to implement the recommended architecture ☑ Develop the software architecture for the system ☑ Develop build-to specifications	☑ Lack of a specification to build the system ☑ Are there alternatives that were missed? ☑ Lack of a modular design ☑ Lack of unit verification ☑ Lack of documentation for legacy system to make the needed changes to the affected areas of the system ☑ Not all requirements are addressed	**Provide Content to SEMP** Development Plan and schedules Configuration Management Plan Risk Plan Integration Plan Deployment Plan Security Plan **Definition on how to implement the recommended architecture:** Detailed design of software architecture Specify the internal interfaces between the central system software for new functionality Specify Java applets developed at the host for remote access Detailed design specifications (code-to) for the Java applets and user interface Specify VPN strategy Detailed design of Oracle application Specify an internet server using Apache technology and Oracle server Specify a T1 communications link with ISP Design data tables and schemas

Process Step	Estimated Level of Effort	Check list of supporting activities	Check list of issues	Check list of risks	Examples
			☑ Establish a detailed design baseline ☑ Verification of the units ☑ Identification of configuration items ☑ Critical design review prior to implementation ☑ Define and document the development environment ☑ Prototype user interface ☑ Traceability between detailed design and requirements	☑ Losing configuration control	
Hardware and software development	Defined by SEMP – Development Plan	☑ Technical reviews ☑ Config mgt ☑ Risk mgt	☑ Development of software ☑ Purchase of commercial off-the-shelf products ☑ Development of COTS application software ☑ Configuration management of software at the developmental level ☑ Code walk-through ☑ Start development of	☑ Lack of traceability between the coding and the detailed design documentation ☑ Not implementing all functionality requirements ☑ Not meeting performance requirements ☑ Losing configuration	**Development of Software** Coding of individual units of software Coding libraries Checking in and checking out of software for CM Code data tables **Purchase of COTS products** Software license Maintenance contracts Communications links

Process Step	Estimated Level of Effort	Check list of supporting activities	Check list of issues	Check list of risks	Examples
			user documentation ☑ Develop Unit Verification Procedures ☑ Developmental engineering reviews ☑ Identify project risks	control	
Unit verification	Defined by the SEMP Development Plan	☑ Technical Reviews ☑ Config mgt ☑ Risk mgt	☑ Check out of the units of software and hardware ☑ Communications ☑ COTS products and applications ☑ Traceability to detailed design ☑ Complete unit functionality	☑ Propagating defects to a higher level ☑ Inability to verify units and to complete development at the unit level	**Check out the units of software and hardware** Check out purchased servers Integrate basic COTS applications with server and verify operations Check units of software that it can perform as specified **Installed communications check** End-to-end test (Pinging messages) Evaluate data rates and delays
Unit integration	Defined by SEMP Integration Plan	☑ Technical review – verification readiness review ☑ Config mgt ☑ Risk mgt	☑ Integrate units of software into sub-systems ☑ Develop Sub-system Verification Procedures	☑ Interfaces are not compatible ☑ Propagating defects to a higher level	**Integrate units of software into sub-systems** Application software for Oracle into the server Integrate Apache application with internet server
Sub-system verification	Defined by the SEMP Verification Master Plan and Verification Procedures	☑ Technical review – verification readiness review ☑ Config mgt ☑ Risk mgt	☑ Verification of sub-systems for functionality ☑ Make ready for the next level of integration ☑ Update user	☑ Interfaces are not compatible ☑ Sub-system functions not complete ☑ Not meeting performance	**Verification of sub-systems for functionality** Verify that the database management system is functional and that the data tables are populated and can be accessed within the performance requirements The Apache application is functional and accessibility of the server to the internet is functional

Process Step	Estimated Level of Effort	Check list of supporting activities	Check list of issues	Check list of risks	Examples
			documentation ☑ Complete sub-system functionality	☑ Propagating defects to a higher level	
Sub-system integration	Defined by the SEMP Integration Plan	☑ Technical review ☑ Verification readiness review ☑ Config mgt ☑ Risk mgt	☑ Integrate sub-systems together into the final system configuration ☑ Update user documentation ☑ Update Operations and Maintenance Plans ☑ Initial deployment and transition into Operations Plan update ☑ System Verification Procedures	☑ Operating agency staff not ready for operations and maintenance ☑ Final configuration is not appropriate ☑ Documentation for uses is not ready ☑ Propagating defects to a higher level	**Integrate sub-systems into the final systems configuration** Integrate the Apache server and internet communications with the Java applet exercise system, end-to-end check for memory leaks, fault conditions, browser compatibility, security, sign filtering (be able to access only the signs required for agency B). Check Oracle database for agency profiles and login authority.
System verification	Defined by SEMP Verification Plan	☑ Stakeholder involvement ☑ Technical reviews ☑ Verification readiness reviews ☑ Config/ Risk mgt	☑ Verify that the system meets all requirements ☑ All documentation is updated and ready for users ☑ Complete system functionality	☑ System not fully implemented ☑ System does not meet requirements ☑ System does not meet the needs of the user	**All documentation is updated and ready for users** All user training, maintenance, user manuals are completed
Deployment	Defined by SEMP Deployment Plan	☑ Technical review ☑ Deployment readiness review	☑ Determine if system is ready to be deployed ☑ Evaluation period ☑ Training updates	☑ System is not ready to be deployed ☑ Latent defects that did not surface during	**Determine if system is ready to be deployed** Staff is trained, Internet access is available, VPN is configured, Agency profiles are fully populated, access to the correct signs have been verified, remote users can read the 6 CMS status and post appropriate messages

Process Step	Estimated Level of Effort	Check list of supporting activities	Check list of issues	Check list of risks	Examples
				verification	
Validation	Defined by SEMP Validation	☑ Stakeholder involvement ☑ Risk mgt ☑ Config mgt	☑ Pre-system studies vs. post-system evaluation, effects on event management	☑ Not meeting the expectations of the stakeholders ☑ Not meeting the needs as specified in Concept of Operations	**Pre-system studies vs. post-system evaluation, effects on event management** In the pre-system evaluation it took 7 staff and 2 hours to set up the event management process; the effects on the event - 30 minutes from the end of the event to move traffic out of the area. It took 45 minutes prior to the event to park the event attendees In the post-system evaluation it took 1 staff and 10 minutes to set up the event management process, and the effectiveness of dynamically changing the signs shows that it took only 15 minutes to clear the event and 30 minutes to park the event attendees.
Operations and maintenance	Defined by the Operations and Maintenance Plan	☑ Stakeholder involvement ☑ Config mgt	☑ On-going maintenance costs ☑ On-call services contracts with COTS vendors ☑ IT support for VPN and internet access	☑ Lack of maintenance ☑ Lack of vendor support	**On-call services contracts with COTS vendors** Updates to Oracle, notice of obsolescence, design changes
Changes and upgrades	Defined by new project SEMP	☑ Stakeholder involvement ☑ Config mgt	☑ Other agencies want access to signs in their jurisdictions	☑ Need for new development ☑ Locked into a specific development team	**Other agencies want access to signs in their jurisdictions** Since the sharing control sub-system was designed for flexibility, it was found that no new development was needed, that adding new profiles and VPNs for the participating agencies would allow the system to accommodate new users without further design Since the new functionality was well documented, the agency has a choice of future development teams and additional functionality, if needed, or to do it themselves.

7.3.3 Example Project 3 - Implementing a New Central Management System

(Please note that the solution given here is for this example only. Other viable solutions may be possible and each must be evaluated for a given project)

This project involves replacing an obsolete traffic signal management system with a new system. This system uses computers at City Hall to provide remote monitoring and control of traffic signals. The existing system is no longer supported by the manufacturer, is unreliable and not maintainable, and lacks functionality needed and available in more modern systems. A needs and feasibility study identified needs and high-level requirements, investigated the capabilities and costs of available off-the-shelf traffic signal management systems, and concluded that an off-the-shelf system using existing communications infrastructure will suffice, but that existing signal controllers (not cabinets) will need to be replaced in addition to central computers and software. The estimated project cost is $1,500,000 exclusive of on-going operation and maintenance costs. There is no immediate need for center-to-center interaction between this system and other systems, but support for data exchange in the future is desirable.

This is a relatively straight-forward project that requires only a low to moderate level of systems engineering. The absence of new software development, use of off-the-shelf components, re-use of existing communications infrastructure, and absence of integration with other systems all reduce risk and complexity. On the other hand, selection of the optimum system, ensuring the new system operates effectively, achieving a smooth transition from the old system to the new, ensuring operations and maintenance personnel are adequately trained, and having the project completed within budget and on schedule will require careful project management including appropriate systems engineering.

Process Step	Estimated Level of Effort	Check list of supporting activities	Check list of issues	Check list of risks	Examples
Feasibility	Medium 2-5 pages	☑ Products survey ☑ Identification of alternative approaches ☑ High-level trade study ☑ Stakeholder involvement (personnel from management, operations, maintenance) ☑ Risk management	☑ Documentation and analysis of: ☑ Needs to be addressed ☑ Possible solution concepts ☑ Estimated cost and benefit ☑ Feasibility with off-the-shelf systems ☑ Project risks ☑ Compatibility with regional ITS architecture ☑ Technical metrics and performance measures	☑ Picking a point solution without considering the business case or cost benefit of alternatives. ☑ Proposing a solution that is too costly ☑ Incompatible components	**Definition of the need:** "Provide a traffic signal management system that is reliable, maintainable, and provides the needed functionality". **Feasibility:** Do available off-the-shelf products address the need? How much of the existing equipment and infrastructure will need to be replaced or upgraded? Is there an affordable solution? **Trade study and cost benefit:** Evaluate alternative approaches, such as retaining existing computers, controllers, cabinets, and communications infrastructure versus replacing some or all of these. Consider alternative communications protocols and their impact on initial and future product choices and communications infrastructure requirements and options. **Stakeholder issues:** Consider performance measurement needs of management, monitoring and control features needed by operations personnel, and self-diagnostic features needed by maintenance personnel. **Risk management:** Ask product vendors for input and cost estimates to ensure proposed solution is feasible and affordable.

Process Step	Estimated Level of Effort	Check list of supporting activities	Check list of issues	Check list of risks	Examples
					Base analysis on mature, proven products. Forego requirements that would require modification to off-the-shelf software or hardware. Give preference to flexible, standards-based solutions.
Planning	Medium 3-5 pages Systems Engineering Management Plan	☑ Describe the work tasks including project management ☑ Identify project execution team, its organization, and the role of each member ☑ Identify any consultant, system integrator, or vendor contracts needed ☑ Document estimated cost and funding sources ☑ Prepare a project time schedule ☑ Identify needed Systems Engineering Plans and their outlines	☑ Availability and expertise of in-house staff ☑ Effort and time required to get consultants on board ☑ Need for independent verification (acceptance testing) ☑ Allowance for contingencies in budget and schedule ☑ Project management techniques and tools to be used ☑ Consider need for on-going maintenance contract with vendor	☑ Loss of control of the project deliverable, schedule and budget ☑ Missing critical activities ☑ Personnel changes	The identified systems engineering plans might include: Deployment Plan Verification and Validation Plans Project Plan Operations and Maintenance Plan Configuration Management Plan Risk Management Plan
Concept of Operations and Validation Plan	Low pages Traffic signal	☑ Elicitation ☑ Stakeholder involvement	☑ Description of the way the system will operate and be	☑ Lack of understanding on: ☑ How the system will be operated	**How the system will operate:** Central software will continuously monitor the operation of traffic signals, reporting current status, traffic flow data, and alarms. The system automatically synchronizes the clocks in signal controllers and commands controllers to change timing

Process Step	Estimated Level of Effort	Check list of supporting activities	Check list of issues	Check list of risks	Examples
	management systems are well understood and do not need an elaborate concept of operations or Validation Plan	☑ Risk management ☑ Trade studies ☑ Technical reviews	maintained ☑ Identification of the project level stakeholders e.g. ☑ Maintenance ☑ Operators ☑ Managers ☑ IT department ☑ Definition of user needs at the project level ☑ Agency's normal operations and maintenance standards ☑ Project goals and measures of effectiveness (for validation)	or maintained ☑ Scope of the project ☑ What functionality is available in off-the-shelf products	patterns when appropriate. Operators periodically check status, update signal timings, respond to alarms, use collected data in various analyses, add new signals to the system, and use the system to temporarily adjust signal timings during incidents. **Measures of effectiveness for validation:** Traffic signal equipment failures are reliably detected and appropriate personnel notified in a timely manner. Data collected by the system are successfully used for traffic analysis and signal timing refinement. Adjusted signal timings are downloaded reliably when needed. The system is used to temporarily adjust signal timings during incidents. Controller clocks are kept synchronized. System users give favorable reports as to the ease of use and effectiveness of the system. The system has a low failure rate and does not require an unreasonable amount of maintenance. The number of citizen complaints that could be avoided by an effective system is reduced.
Development of System Level Requirements and Verification Plans	Low - Medium 2-4 pages Requirements and 2-4 page Verification Plan	☑ Stakeholder involvement ☑ Elicitation ☑ Technical reviews ☑ Trade studies ☑ Risk management ☑ Configuration management ☑ Traceability	☑ Identification of system requirements to support the identified needs ☑ What will be used as the basis for accepting the installed system? ☑ New risks that may be uncovered ☑ Are all the needs	☑ Not having a basis to accept the system when completed ☑ Not completely defining what the system is to do ☑ Scope creep ☑ Expectations not met ☑ Project cost ☑ Losing control	**Definition of what the system is to do to support the identified needs.** "The traffic signal system shall automatically synchronize controller clocks at a user-selectable time of day." "The system shall allow users to upload and store controller data sets." "The system's central software shall operate on personal computers using the Windows operating system." "The system shall provide three workstations.". **What will be used for the basis of verification and acceptance of the system?** Verification Plan would include, for example: Configure a clock update time in the near future and check that the test controller's clock changes to match the server's

Process Step	Estimated Level of Effort	Check list of supporting activities	Check list of issues	Check list of risks	Examples
			addressed and is each need addressed completely? ☑ What will be needed to support operations and maintenance? ☑ Project cost update	and visibility of the development ☑ Requirements not validated by stakeholders	clock at that time. Upload a designated controller's data set, store the data, restart the database server, check that the stored data are available and match that in the controller. **What will be needed to support operations and maintenance?** Users and maintenance documentation shall be provided. System configuration documentation shall be provided.
System design	Low - medium 5-8 pages Design Document including map and diagrams	☑ Stakeholder involvement, ☑ Trade studies ☑ Risk management ☑ Configuration management ☑ Technical reviews ☑ Procurement	☑ Evaluate alternative off-the-shelf systems by comparing against requirements and considering costs ☑ Procure the preferred system ☑ Work with supplier to prepare system design ☑ Project costs and risks ☑ Refine Verification Plan	☑ Not deployable ☑ Not maintainable ☑ Inflexible ☑ Lack of standards ☑ Project costs	**Examples of design elements:** including any cabinet wiring changes, controller options, communication protocol choice, computer furniture, graphics style. Map showing location of signals to be integrated in the new system and communication infrastructure used. Any cabinet wiring modifications. Controller firmware version to be installed. Process for converting and transferring existing controller data sets to the new controllers. Any new racks and furniture needed at central. Configuration of computers. Graphics style and source of base drawings. Naming conventions. Definition of signal groupings. Cutover Plan. Training Plan. **Project costs** Revised cost estimate based on final system design
Development of Component	No detailed component-level				

Process Step	Estimated Level of Effort	Check list of supporting activities	Check list of issues	Check list of risks	Examples
Level Design	design needed since all system components are off-the-shelf.				
Hardware and software development	Not needed, since all components are off-the-shelf				
Unit verification	Defined by the Deployment Plan and Verification Plan	☑ Technical reviews ☑ Configuration management ☑ Risk management	☑ Inspect and test the units of software and hardware ☑ Traceability to detailed design ☑ Complete unit functionality	☑ Propagating defects to a higher level ☑ Inability to verify units	**Inspect and test the units of software and hardware** Computers and installed software – before any signals are connected Controllers and installed firmware – stand alone bench tests
Unit integration	Defined by the Deployment Plan	☑ Technical review – verification readiness review ☑ Configuration management ☑ Risk management	☑ Integrate units of software into sub-systems ☑ Develop Sub-system Verification Procedures	☑ Interfaces are not compatible ☑ Propagating defects to a higher level	**Integrate components into sub-systems** Prepare and install a test controller in a bench cabinet. Connect the bench controller in a cabinet to the communications server Install software and data sets in all controllers, ready for installation
Sub-system verification	Defined by the Verification Plan and	☑ Technical review – verification	☑ Verification of sub-systems for	☑ Interfaces are not compatible	**Verification of sub-systems for functionality** Verify that the central software can successfully

Process Step	Estimated Level of Effort	Check list of supporting activities	Check list of issues	Check list of risks	Examples
	Verification Procedures	readiness review ☑ Configuration management ☑ Risk management	functionality ☑ Make ready for the next level of integration ☑ Update user documentation	☑ Sub-system functions not complete ☑ Not meeting performance ☑ Propagating defects to a higher level	communicate with a test controller on the bench. Test system functionality with the bench controller connected.
Sub-system integration	Defined by the Deployment Plan	☑ Technical review ☑ Verification readiness review ☑ Configuration management ☑ Risk management	☑ Integrate sub-systems together into the final system configuration ☑ Update user documentation ☑ Update Operations and Maintenance Plans ☑ Initial deployment and transition into Operations Plan update ☑ System verification procedures	☑ Operating agency staff not ready for operations and maintenance ☑ Final configuration is not appropriate ☑ Documentation for uses is not ready ☑ Propagating defects to a higher level	**Integrate sub-systems into the final system's configuration** Make any needed modifications to cabinets in the field. Install new controllers in the field. Complete installation of computers and other central equipment. Connect field controllers to the central communications server. Complete central software configuration.
System verification	Defined by the Verification Plan	☑ Stakeholder involvement	☑ Verify that the system meets all	☑ System not fully implemented	**System passes all acceptance tests.** Perform system-level acceptance tests in accordance with the Verification Plan.

Process Step	Estimated Level of Effort	Check list of supporting activities	Check list of issues	Check list of risks	Examples
		☑ Technical reviews ☑ Verification readiness reviews ☑ Configuration management ☑ Risk management	requirements ☑ All documentation is updated and ready for users ☑ Complete system functionality	☑ System does not meet requirements ☑ System does not meet the needs of the user	**All documentation is updated and ready for users** All user training, maintenance, user manuals are completed. The system configuration is fully documented.
Deployment	Since this is a replacement system, deployment has already occurred by this stage.			☑ New system cannot be co-installed with old system ☑ New system cutover without a fall back system	
Validation	Defined by SEMP Validation Plan	☑ Stakeholder involvement	☑ Evaluate system effectiveness	☑ Not meeting the expectations of the stakeholders ☑ Not meeting the needs as specified in Concept of Operations	**System evaluation.** Does the system provide the benefits expected? Are users able to use it effectively? Are field equipment faults reported reliably and quickly? Is it reliable and easy to maintain?
Operations and maintenance	Defined by the Operations and Maintenance	☑ Stakeholder involvement	☑ On-going maintenance	☑ Lack of maintenance	**On-going operation and maintenance.** Ensure operations staff are available and using the system as needed.

Process Step	Estimated Level of Effort	Check list of supporting activities	Check list of issues	Check list of risks	Examples
	Plan	☑ Configuration management	costs ☑ On-call service contracts with vendors ☑ IT support for servers and remote access	☑ Lack of vendor support	Arrange vendor support contracts if needed after warranty period. Provide operation and maintenance training for new employees. Maintain spare parts inventory. Keep system configuration documentation up to date. Track system shortcomings and additional needs. Plan changes and upgrades when needed.
Changes and upgrades	Defined by new project SEMP	☑ Stakeholder involvement ☑ Configuration management	☑ System not performing needed functions ☑ System becoming difficult to maintain.	☑ Need for new software development ☑ Locked into a specific vendor	**Examples of reasons to change or upgrade** Equipment or software may become obsolete May need to add a center-to-center connection May need to add cameras or signs.

7.4 Contract Guidance

This section provides general guidance on two aspects of contracting, Request for Proposal, and the Intellectual Property Rights Clause. These are especially important and are somewhat unique for Intelligent Transportation Systems. However, it is critical to note that the general guidance contained here is not meant to supersede guidance from your own agency. Project developers must always check with their agency's contracting and procurement staff for specific and mandatory guidance on contracting issues.

Most aspects of contracting, or procurement, are the same across almost all categories of the transportation infrastructure. Indeed, some are common to almost any type of procurement. However, any development of a software-intensive ITS brings up a whole host of new considerations, considerations that stretch all the way from initial concept, through requirements, design, build and verification. Since it is very common to contract for some of these efforts, with consultants and system developers, some of these special considerations can reflect back into the contracting documents and processes themselves.

Although this Guidebook touches on procurement, it is only as it relates to the systems engineering processes. Other documents, available from the Federal Highway Administration, will provide much more information on the procurement options for Intelligent Transportation Systems.

In this section, only two aspects of procurement are discussed. The first is guidance on the contents of a Request for Proposal package. The second is the need for an Intellectual Property Rights clause to give the procuring agency the rights it needs for follow-on maintenance and upgrades of software products.

This section contains guidance on the following topics:

- Request for Proposal
- Intellectual Property Rights Clause

7.4.1 Request for Proposal Guidelines

Preparation of a Request for Proposal, or RFP, is very common when developing an Intelligent Transportation System. This section discuses the technical and project management aspects of an RFP. It does not cover, with the exception of the following section on Intellectual Property Rights, contractual or legal issues related to procurement. Procurement options are covered in Section 4.8.7.

The primary purpose of an RFP is to tell prospective contractors what it is you want. It is always well to remember that if you don't ask for it, you probably won't get it. This is especially true when the procurement is going to be cost competitive. Here, a contractor is going to be penalized if they provide something you need but didn't specifically ask for. This also will cause difficulty in justifying their higher price. On the other hand, if you are too specific the contractor may be discouraged from bidding a lower cost solution that is perfectly adequate.

Following is a list, and description, of the most commonly found parts of a good RFP. Not all of these parts may be required every time and for every type of procurement.

Statement of Work (or Project Plan)

The Statement of Work, or Project Plan, is where you tell the potential contractor what it is that you want them to do. The Statement of Work is very much like the Project Plan discussed in sections 4.3.1, Project Planning, and 7.5.1, Project Plan Guidelines. It always must contain a high level description of the project to give the prospective contractors a sense of the context of their work. It is generally written by breaking down the work to be done into a number of individual tasks. Then each task is described by its inputs, approach and outputs. The inputs are the information needed by the contractor to do the task. The approach is guidance on how the task is to be done (for instance, specific trade-off studies may be required). The outputs are the products of the task, such as, documents, meetings and deliverable hardware and software. It is especially important to describe the contents of any required documents. Section 7.5 of this Guidebook gives descriptions of some of the most common documents that could be required. All tasks, including administrative tasks, need to be included in the Statement of Work.

Technical Specifications

One or more technical specifications may be required to describe the products to be procured. If the product is off-the-shelf, a part number or model number is all that is necessary. If the product is to be developed, then a requirements specification is needed. Section 7.5.6 provides guidelines for a requirements specification.

Sometimes the requirement specifications are planned to be written as part of the contractor's effort. In this case, some level of a description of the intended product is still needed and this must be prepared before the RFP is released.

Schedule

The RFP must contain a schedule for the tasks. This schedule should be at as high a level as possible and only contain dates that must be met by the contractor. Examples include a start date, a delivery date(s), or a date after which an important external system is available. Let the contractor propose back to you internal dates, such as delivery of a specification, delivery of design documents and start of verification. In fact, the contractor should propose a schedule for every deliverable of every task.

List of Deliverables

It is good practice to compile a comprehensive list of all deliverables, including documents, products and meetings. This identifies, in one location, all of the important evidence you will have that a task is satisfactorily completed. All deliverables are referenced to the task that produces them (they are also listed as outputs of that task). Documents are referenced to the document descriptions and products are referenced to the appropriate specification, if any. This list includes information about quantities. It also includes ground rules for agency review and comment and for incorporation of comments.

Contract Terms and Conditions

Contract terms and conditions are generally provided by the agency's procurement or purchasing departments. Much of these contract terms are standardized and apply to any procurement. However, some will be specialized to a specific type of contract or procurement and it is important that they are compared to other parts of the RFP package to avoid conflicts.

Format of Proposal

The RFP needs to tell prospective bidders what their response should look like. Generally, this is in terms of both form and content. Quite often, for instance, a page limit is placed on their response, which helps them keep their focus on the items the agency wants to see. The main purpose of specifying the format and content of the response is to make it easier to compare bids from different contractors and to ensure enough information is available to make that comparison.

A typical proposal may be asked to include:

- Certifications by the contractor, for instance, certification of a clean record with other agencies or certification of meeting a Disadvantaged Business Enterprise requirement.
- Demonstration of compliance with the RFP, including a complete list of any areas of known non-compliance. However, optional responses, for instance, a non-compliant, but lower cost way of meeting a requirement, may be allowed.
- A preliminary Project Plan, perhaps providing some detail on their approach to be used for each task. This part of the response is critical in determining and evaluating the contractor's understanding of the project.
- A preliminary system design.

In addition, an RFP always asks for:

Qualification and Experience

It is always necessary to obtain information about the qualifications and experience of the contractor and its key employees. This provides an indication that the contractor can perform the work requested. The demonstration of experience should include a list of comparable projects completed and references to the procuring agency. Qualification of employees is often provided in tailored resumes.

Cost Breakdown

Quite often, the RFP requests that all cost data be submitted separately from the rest of the response. This is done so that the technical evaluation of the proposal is not tainted by knowledge of the cost of the proposal.

The cost figures are generally requested to be broken down in several different ways. Typically, costs are broken down by task. Within a task, the breakdown between labor costs and other direct costs (itemized) are shown. Labor is also shown by hours and by labor category (e.g. senior engineer, junior engineer).

Sometimes, a breakdown of the total cost is requested to show labor costs, direct costs, overhead, and fee.

RFP Evaluation Criteria

To complete the RFP, guidance is given to the contractors on how the proposals will be evaluated. It is rare that an ITS proposal is evaluated by cost alone. Generally technical, management, experience and qualifications are given equal, if not a higher, weighting. The purpose of this part of the RFP is to show contractors exactly what factors (based on information contained in their responses) will be evaluated and what relative weights will be given to each.

7.4.2 Intellectual Property Rights Guidelines

One of the most challenging contractual problems associated with Intelligent Transportation Systems is the establishment of adequate rights for the agency with respect to the system's software. It has not been unusual for agencies to pay for the development of some custom transportation software programs only to find, because of restrictive contract language, they can not even see the code. The agency also may find that they must go back to the original developer for maintenance and for future system upgrades.

The rights of the agency to use a software product are established in an Intellectual Property Rights clause that is put into the terms and conditions of their contract with the software developer. There are three aspects to these rights that are generally covered in this clause. The first is the agency's right to receive from the developer not only the executable code, but also the source code and the related documentation needed to understand and to replicate that code. The second is the agency's right to use the code and the documentation to maintain and upgrade the software program, including the right to give the software and documentation to another development contractor for maintenance or upgrade. The third is the necessity to recognize that the original developer has some rights to the software as well, especially if they, and not the agency, paid for development of it.

To illustrate these points, example wording for the parts of an Intellectual Property Right clause is derived from wording developed by the Department of General Services of the State of California. This example is not intended to be used in a contract as an Intellectual Property Rights clause. A project manager should contact his agency's legal staff for appropriate contract wording.

Issue 1 – Getting the Necessary Documentation

This is really a problem for the Contract's Statement of Work. The Statement of Work is where the deliverable documents and products are defined, and where the delivery of executable and source code as well as other related documents is defined. Related documents may include High Level and Detailed Design Specifications, Software Development Plans, instructions on how to compile the executable code, and user's manuals.

A sample clause for the contract is:

> ***DOCUMENTATION***
>
> *The Contractor agrees to provide to the Agency a number of all nonproprietary manuals and other printed material, as described with the Statement of Work, and updated versions thereof, which are necessary or useful to the Agency in its use of the Equipment or Software provided hereunder.*

Issue 2 – Getting the Right to Use the Software and Related Documentation

This part of the Intellectual Property Rights puzzle is handled entirely in the Contract. Here the agency wants to get and retain the unencumbered rights to:

- Use the software in the system it was designed for
- Give the software and documents to a third party (another contractor) for maintenance and for upgrades
- Give the software and related documents to another government agency for their use

The example clause does this by defining a set of "Government Purpose Rights" which it then gives to the agency. In doing this, it also defines rights which are left as the sole or joint right of the development contractor.

A sample clause for the contract is:

> ***RIGHTS IN WORK PRODUCTS***

a) All inventions, discoveries, intellectual property, technical communications and records originated or prepared by the Contractor pursuant to this Contract including papers, reports, charts, computer programs, and other Documentation of improvements thereto, and including Contractor's administrative communications and records related to this Contract (collectively, the "Work Product"), shall be Contractor's exclusive property. The provisions of this sub-section may be revised in a Statement of Work.

b) Software and other materials developed or otherwise obtained by or for Contractor or its affiliates independently of this Contract *or applicable purchase order ("Pre-Existing Materials") do not constitute Work Product. If Contractor creates derivative works of Pre-Existing Materials, the elements of such derivative works created pursuant to this Contract constitute Work Products, but other elements do not. Nothing in this Section will be construed to interfere with Contractor's or its affiliates' ownership of Pre-Existing Materials.*

c) *The Agency will have Government Purpose Rights to the Work Product as Deliverable or delivered to the Agency hereunder. "Government Purpose Rights" are the unlimited, irrevocable, worldwide, perpetual, royalty-free, non-exclusive rights and licenses to use, modify, reproduce, perform release, display, create derivative works from, and disclose the Work Product. "Government Purpose Rights" also include the right to release or disclose the Work Product outside the Agency for any Agency purpose and to authorize recipients to use, modify, reproduce, perform, release, display, create derivative works from, and disclose the Work Product for any Agency purpose. Such recipients of the Work Product may include, without limitation, Agency Contractors or other agencies. "Government Purpose Rights" do not include any rights to use, modify, reproduce, perform, release, display, create derivative works from, or disclose the Work Product for any commercial purpose.*

d) *The ideas, concepts, know-how, or techniques relating to data processing, developed during the course of this Contract by the Contractor or jointly by the Contractor and the Agency, may be used by either party without obligation of notice or accounting.*

e) *This Contract shall not preclude the Contractor from developing materials outside this Contract that are competitive, irrespective of their similarity to materials which might be delivered to the Agency pursuant to this Contract.*

Issue 3 – Agreeing to the Rights of the Development Contractor

The third and final issue that the Intellectual Property Rights clause must deal with is recognizing rights retained by the developer. Not all the software components the developer uses in the system software were developed just for this project. Sometimes, the developer uses software that they have developed, at their own expense, as a product line. Because the agency does not pay their development costs, they cannot expect to get the same all-inclusive rights as they would get to custom software. In addition, many components of the software, such as the operating system, a database engine, or communications packages, have been developed by third parties. Since the developer's business livelihood is dependent on the exclusive ownership of those products, the agency not only gets limited rights, but also must take reasonable steps to protect what information they do get from the original developer's competitors.

A sample clause for the contract is:

PROTECTION OF PROPRIETARY SOFTWARE AND OTHER PROPRIETARY DATA

a) *Agency agrees that all material appropriately marked or identified in writing as proprietary, and furnished hereunder are provided for Agency's exclusive use for the purposes of this Contract only. All such proprietary data shall remain the property of the Contractor. Agency agrees to take all reasonable steps to insure that such proprietary data are not disclosed to others, without prior written consent of the Contractor, subject to law.*

b) *The Agency will insure, prior to disposing of any media, that any licensed material contained thereon have been erased or otherwise destroyed.*

c) *The Agency agrees that it will take appropriate action by instruction, agreement or otherwise with its employees or other persons permitted access to licensed software and other proprietary data to satisfy its obligations under this Contract with respect to use, copying, modification, protection and security of proprietary software and other proprietary data.*

Use of software escrow accounts is one approach that allows the contractor to maintain rights and also protects the buyer. Software escrow account is a specialized firm that holds intellectual property such as sources code and design documentation in case the contractor defaults, or goes bankrupt then it is released to the buyer.

7.5 Systems Engineering Documentation Guidance

This section provides guidance on the preparation of some of the most commonly used systems engineering documents. This guidance is in the form of a preliminary discussion of the purpose of the document, suggestions on tailoring the document to the project, and a checklist of critical information to be included. This is followed by a suggested document outline showing and explaining each major part or section of the document.

Each document is referenced back to its process section in chapter 4. For best effect, the process sections and the document guidance sections should be studied together.

- Guidance is provided on the following systems engineering documents:
- Project Plan (see process section 4.3.1, Project Planning)
- Systems Engineering Management Plan (see process section 4.3.2, Systems Engineering Management Planning)
- Configuration Management Plan (see process section 4.8.6, Configuration Management / Interface Management)
- Needs Assessment (see process section 4.2.2, Needs Assessment)
- Concept of Operations (see process section 4.3.3, Concept of Operations)
- Requirements Specification (see process section 4.4.1, Requirements Development)
- Design Specifications (see process sections 4.4.2, High Level Design and 4.4.3, Component Detailed Design)
- Integration Plan (see process section 4.5.2, Integration)
- Verification Documents (see process section 4.5.3, Verification)
- Deployment Plan (see process section 4.5.4, Initial System Deployment)
- Operations and Maintenance Plan (see process section 4.6.2, Operations and Maintenance

7.5.1 Project Plan Guidelines

Purpose of this Document

The Project Plan is the governing document for conduct of your project. All other plans and technical documents follow from the Project Plan. Most agencies have project management procedures that call for the creation of a Project Plan. Obviously, those need to be followed. The Project Plan described here shows the most commonly needed elements of a Project Plan.

The purpose of the Project Plan is to define and describe all the tasks that need to be performed to accomplish the project. Each task is described in enough detail that the assigned personnel can do it satisfactorily. It is also critical that the products of each task, the schedule for each task, and the available budget are established. Further, the assigned personnel need to "buy-in" to this plan and believe that they can do their task on time and within budget.

Also the Project Plan establishes and identifies the environment in which the project will operate. It identifies all the players in the project including management, responsible teams or organizations for each task, supporting organizations, and all stakeholders.

Tailoring This Document To Your Project

Although almost always required, the size of the Project Plan can vary considerably depending on the complexity of the project and the breadth of its environment. If needed, the Project Plan can be supplemented with a variety of supporting plans. Depending on complexity, it may be more efficient to document all this support plan information in the Project Plan itself, or at least in no more that one or two other plan documents.

Among the complexity factors that affect the depth of the Project Plan are:

Cost of the project. The more expensive a project, the more that your management will want to see that it is planned well.

The technical complexity of a project. Technical complexity translates directly into technical risk that must be managed through good planning.

The number of involved stakeholders. These stakeholders will use the Project Plan to understand and plan their roles and it provides a means for them to review and comment on their ability to perform the needed tasks.

Checklist Of Critical Information

- ☑ Are all of the necessary tasks included in the plan (perhaps in the form or a Work Breakdown Structure) along with identification of the personnel or team that is responsible for performing the task?
- ☑ Is the sequence of the tasks correct so that the necessary precursor work is done for each task?
- ☑ Is the budget assigned to each task sufficient to get the task done as defined, and does the team that will perform the task agree?
- ☑ Is the scheduled time period for each task sufficient to get the work done as defined, and does the team that will perform the task agree?
- ☑ Are the necessary stakeholder organizations identified and their roles defined, and agreed to?
- ☑ Are all products of each task (documents, meetings, hardware and software) identified, or alternatively, is a task defined to identify those products?
- ☑ Are any supporting plans required to supplement the Project Plan, and is their preparation defined as a task?
- ☑ Do all stakeholders, including your management, approve the Project Plan?

PROJECT PLAN GUIDELINES

SECTION	CONTENTS
Title Page	The title page should follow the Transportation Agency procedures or style guide. As a minimum, it should contain the following information: ▪ PROJECT PLAN FOR THE Name of Project AND Transportation Agency ▪ Contract number ▪ Date the document was formally approved ▪ The organization responsible for preparing the document ▪ Internal document control number, if available ▪ Revision version and date issued
1.0 Purpose of Document	A brief statement of the purpose of this document, that is, the plan for execution of the project defining all necessary tasks and their products.
2.0 Scope of Project	This section gives a brief description of the planned project and the purpose of the system to be built. Special emphasis is placed on the project's complexities and challenges to be addressed by the project's managers. This section defines the project's relationship to the applicable regional ITS architectures and, if necessary, to the National ITS Architecture. It also defines the relationship of the project's system to other systems with which it interfaces, either physically (with a data interface) or operationally. This section describes the environment in which the project will operate. It identifies the organizational structures that encompass all stakeholders and gives a brief description of the role to be played by each stakeholder. This section identifies organizations within the owning agency that are stakeholders in this project as well as any external agencies (especially agencies with a system that interface with this project's system) that are project stakeholders. A subsequent project management task is to identify individuals within those organizations and agencies who will represent their organization among the project's stakeholders. It is especially important that the Project Plan identify the system owners who are building the system and the customer for whom the system is being built. The section also identifies any existing the management work groups and multi-disciplinary technical teams to be used to support the project.

3.0 Project Tasks	This section is the heart of the Project Plan. It defines each task of the project in terms of its inputs, its approach, and its outputs.
	▪ **Inputs** – Identification of the inputs to each task. Inputs can be a variety of things, including, but certainly not limited to:
	  ▪ Documents, from outside the project or from other tasks of the project, that are meant to guide the activities of this task, such as, a regional ITS architecture and other planning documents
	  ▪ Direction from others that guide the efforts of the team performing this task, such as directions from a multi-agency steering committee established for this project
	  ▪ Meetings with others to be conducted by the team performing this task, such as periodic status meetings with the project manager's organizational management.
	  ▪ Products, other than documents, from other tasks that are a necessary precursor for performance of this task. For example, a product from an integration task is a software and hardware component that is a

<table>
<tr><th>SECTION</th><th>CONTENTS</th></tr>
<tr><td></td><td>necessary input to a verification task.
▪ Approach – A description of the approach to be taken by the team performing the task. This may include a description of the products of the task, of the analysis sub-tasks to be undertaken, or even a breakdown of the tasks into sub-tasks. This description may include identification of procurement activities that need to be taken in this task. For systems engineering and design tasks, this description may be expanded as necessary in the Systems Engineering Management Plan, which, of course, would be an activity and output of one of the tasks.
▪ Outputs – A description of the products of the task. As with inputs, the outputs may take many forms, including, but not limited to:
  ▪ Documents to be produced by the task team, such as, specifications, Verifications Plans, the SEMP.
  ▪ Meetings, including management meetings and technical reviews
  ▪ Other products such as software code, procured hardware, integrated or verified sub-systems
  ▪ Attendance at meetings conducted by others, such as periodic meetings of a multi-agency steering committee</td></tr>
<tr><td>4.0 Work Breakdown Structure and Task Budgets</td><td>This section provides a hierarchical structure of all tasks and sub-tasks of the project, identifying the name of the task or sub-task, the allocated budget, and the team or organization with the authorization and responsibility to perform the task. The budget may not be allocated to each sub-task but may be allocated to a higher level group of sub-tasks, tasks, or group of tasks, as necessary to manage the project.</td></tr>
<tr><td>5.0 Schedule Constraints</td><td>A project’s schedule is developed in two steps, and this section, at a minimum, includes information to define the initial step of schedule development. The two steps in development of a project’s schedule are:
▪ Step One -- identification of external schedule constraints. These may include a not-earlier-than start date, a not-later-than completion date, a date tied to the completion of an external system, or the date a needed resource is available. In general, these schedule constraints come from outside the project and are not within the control of the project’s management.
▪ Step Two – development of a schedule for each task, for each sub-task, and for each output of a task. This schedule is under complete control of the project’s management by a variety of means, including assignment of more or fewer of resources. This schedule takes into account the necessary precursors (inputs) to each task or sub-task.
The schedule in this section of the Project Plan includes the output of step one and may either include the complete schedule of step two or identify this as an output of one of the tasks.</td></tr>
<tr><td>6.0 Deliverable Requirements List</td><td>This section is, as much as possible, a complete and precise list of the tangible deliverables of each and every task. In general, a tangible deliverable may include, from the list of outputs of a task:
▪ Documents, especially documents to be reviewed by stakeholders and documents to be used after the system is built.
▪ Meetings and reviews to be attended by project stakeholders
▪ Other products, such as deliverable hardware (by name, part number, and quantity) and deliverable software products, such as source code and</td></tr>
</table>

SECTION	CONTENTS
	executables It may not be possible to completely and precisely define each and every deliverable at the time the Project Plan is prepared. For instance, the Project plan may state that design specifications are required but identification of specific document may have to wait until the sub-systems are defined in the high level design task.
7.0 Referenced Documents	This section lists the applicable documents that are inputs to the project (that is, are needed by but not produced by the project). Such documents may include, the regional ITS architecture description, planning documents describing the project, agency procedures to be followed, standards, and specifications and other descriptions of interfacing external systems. Other applicable documents may be required by a specific project.

7.5.2 Systems Engineering Management Plan Guidelines

Purpose of this Document

The Systems Engineering Management Plan, or SEMP, may be needed to supplement the details of the Project Plan. When used, the SEMP focuses on technical plan of the project and the systems engineering processes to be used for the project. Its purpose is to detail out those engineering tasks and especially to provide detailed information on the processes to be used. Preparation of a SEMP is most important if the project involves development of custom software. The engineering tasks of producing custom software (from requirements, through design implementation, integration and verification) are very complex, and are new to many transportation engineers.

Given the level of process detail needed in the SEMP, it is quite often written in two steps. In the first step, the framework for the document is prepared, usually by the project management staff. Enough detail is included to identify all the needed tasks (including analysis tasks) and any important constraints on the performance of a task (such as use of a specific systems engineering and design methodology). In the second step, the various sections of the SEMP framework are completed, this time by the team that will perform each task. For instance, the requirements team provides details on the analysis and the tools used to manage requirements, the design team provides details on use of the software design methodology, the software coder provides details on configuration management of the software code, and the verification team provides details on their verification methods.

These SEMP guidelines were adapted from guidelines prepared by the Caltrans Office of Local Assistance, see the web page at: http://www.dot.ca.gov/hq/LocalPrograms/lam/prog_g/g12othr.pdf .

Tailoring this Document to Your Project

- The simplest ITS projects may not need a SEMP, finding that the Project Plan is sufficient. Among the project complexities that make preparation of a SEMP desirable are:
- Inexperience of the system owner's project team in the systems engineering tasks and processes
- A larger number of stakeholders and the degree of their involvement in the various systems engineering processes and tasks
- The need to develop custom software applications
- A project where the solution is not well understood and is not generally obvious

Checklist of Critical Information

☑ Are all the technical challenges of the project addressed by the systems engineering processes described in the SEMP?

☑ Does the SEMP describe the processes needed for requirements analysis?

☑ Does the SEMP describe the design processes (and the design analysis steps required for an optimum design)?

☑ Does the SEMP clearly identify any necessary supporting technical plans, such as a Verification Plan or an Integration Plan, and define when and how they will be written?

☑ Does the SEMP spell out stakeholder involvement when it is necessary?

☑ Does the SEMP identify all the required technical staff and development teams and the technical roles to be performed by the system's owner, the project staff, the stakeholders, and the development teams?

☑ Does the SEMP cover the interfaces between the various development teams?

SYSTEMS ENGINEERING MANAGEMENT PLAN GUIDELINES

SECTION	CONTENTS
Title Page	The title page should follow the Transportation Agency procedures or style guide. As a minimum, it should contain the following information: ▪ SYSTEMS ENGINEERING MANAGEMENT PLAN FOR THE Name of Project AND Transportation Agency ▪ Contract number ▪ Date the document was formally approved ▪ The organization responsible for preparing the document ▪ Internal document control number, if available ▪ Revision version and date issued
1.0 Purpose of Document	A brief statement of the purpose of this document, that is, the plan for the systems engineering activities with special emphasis on the engineering challenges of the system to be built.
2.0 Scope of Project	This section gives a brief description of the planned project and the purpose of the system to be built. Special emphasis is placed on the project's complexities and challenges that must be addressed by the systems engineering efforts. This section also describes the environment in which the project will operate. It identifies the organization structures that encompass all stakeholders and gives a brief description of the role to be played by each stakeholder. This includes ad hoc and existing management work groups and multi-disciplinary technical teams that should be formed or used to support the project. Such teams are critical to reaching successful system deployment. This section defines the general process for developing the SEMP, including the draft framework version prepared by the transportation agency and the complete version prepared in conjunction with the development teams.
3.0 Technical Planning and Control	This section lays out the plan for the systems engineering activities. It must be written in close synchronization with the project's Project Plan. Unnecessary duplication between the Project Plan and the SEMP should be avoided. However, it is often necessary to put further expansion on the systems engineering effort into the SEMP even if they are already described at a higher level in the Project Plan. Even within the SEMP, an effort may need to be described at a higher level in the draft SEMP framework and then expanded further in the final version of the SEMP. The purpose of the section is to describe the plans that will act as controls on the project's systems engineering activities. For instance, this section identifies the products of each systems engineering activity, such as, documentation, meetings, and reviews. This list of required products will control the activities of the team performing the activity and will control the satisfactory completion of the activity. Some of these plans may be completely defined in the SEMP (in the framework or the complete version). For other plans, the SEMP may only define the requirements for a particular plan and the plan itself is to be prepared as one of the subsequent systems engineering activities, such as may be the case with a Verification Plan or a Deployment Plan. Almost any of the plans described below may fall into either category. It all depends on the complexity of the particular plan and the amount of up-front systems engineering that can be done at the time the SEMP is prepared. The first set of required plans relates primarily to the successful management

SECTION	CONTENTS
	of the project. These plans are likely to have already been included in the Project Plan, but may need to be expanded here in the SEMP. Generally they are incorporated into the SEMP, but on occasion may be developed as separate documents. ▪ Work Breakdown Structure (WBS) (also included in the Project Plan)– a list of all tasks to be performed on a project, usually broken down to the level of individually budgeted items ▪ Task Inputs – a list of all inputs required for each task in the WBS, such as source requirements documents, interface descriptions, and standards. ▪ Task Deliverables – a list of the required products of each task in the WBS, including documents, software and hardware ▪ Task Control Gates – a list of critical activities that must be satisfactorily completed before a task is considered completed ▪ Reviews and Meetings – a list of all meetings and reviews of each task in the WBS ▪ Task Resources – identification of resources needed for each task in the WBS, including for example, personnel, facilities, and support equipment. ▪ Task Procurement Plan – a list of the procurement activities associated with each task of the WBS, including hardware and software procurement and, most importantly, any contracted services, such as systems engineering services or development services ▪ Critical Technical Objectives – a summary of the plans for achieving any critical technical objectives that may require special systems engineering activities. It may be that a new software algorithm needs to be developed and its performance verified before it can be used. Or a prototyping effort is needed to develop a user-friendly operator interface. Or a number of real-time operating systems need to be evaluated before a procurement selection is made. This type of effort is not needed for all projects. ▪ Systems Engineering Schedule – a schedule of the systems engineering activities that shows the sequencing and duration of these activities. The schedule should show tasks (at least to the level of the WBS), deliverable products, important meetings and reviews and other details needed to control and direct the project. An important management tool, this schedule is used to measure the progress of the various teams working on the project and to highlight work areas that need management intervention. The second set of plans is designed to address specific areas of the systems engineering activities. They may be included entirely in the SEMP or the SEMP may give guidance for their preparation as separate documents. The plans included in the first set listed above are generally universally applicable to any project. On the other hand, some of the plans included in this second set are only rarely required. The unique characteristics of a project will dictate their need. ▪ **Software Development Plan** – describes the organization structure, the facilities, the tools and the processes to be used to produce the project's software. Describes the plan to produce custom software and to procure commercial software products ▪ **Hardware Development Plan** – describes the organization structure, the facilities, the tools and the processes to be used to produce the project's

SECTION	CONTENTS
	hardware. Describes the plan to produce custom hardware (if any) and to procure commercial hardware products ▪ **Technology Plan** – if needed, describes the technical and management process to apply new or untried technology to an ITS use. Generally it addresses performance criteria, assessment of multiple technology solutions, and fall-back options to existing technology. ▪ **Interface Control Plan** – identifies the physical, functional and content characteristics of external interfaces to a system, if any, and identifies the responsibilities of the organizations on both sides of the interface ▪ **Technical Review Plan** – identifies the purpose, timing, place, presenters and attendees, subject, and the entrance criteria (a draft specification completed) and the exit criteria (resolution of all action items) for each technical review to be held for the project ▪ **System Integration Plan** – defines the sequence of activities that will integrate software components into sub-systems and sub-system into entire systems. This plan is especially important if there are many sub-systems and each is produced by a different development team. ▪ **Verification Plan** – almost always required, this plan is written along with the requirements specifications. However, the parts on test conduct can be written earlier. ▪ **Installation Plan** or **Deployment Plan** – describes the sequence in which the parts of the system are installed (deployed). This plan is especially important if there are multiple different installations at multiple sites. A critical part of the deployment strategy is to create and maintain a viable operational capability at each site as the deployment progresses. ▪ **Operations and Maintenance Plan** – defines the actions to be taken to ensure that the system remains operational for its expected lifetime. It defines the maintenance organization and the role of each participant. This plan must cover both hardware and software maintenance. ▪ **Training Plan** – describes the training to be provided for both maintenance and for operation ▪ **Configuration Management Plan** – describes the development team's approach and methods to manage the configuration of the system's products and processes. It will also describe the change control procedures and management of the system's baselines as they evolve. ▪ **Data Management Plan** – describes how and which data will be controlled, the methods of documentation, and where the responsibilities for these processes reside. ▪ **Risk Management Plan** – addresses the processes for identifying, assessing, mitigating, and monitoring the risks expected or encountered during a project's lifecycle. Also identifies the roles and responsibilities of all participating organizations for risk management. ▪ **Other plans** – these might include for example, a Safety Plan, a Security Plan, a Resource Management Plan, and/or Validation Plan. ▪ This second list is extensive and by no means exhaustive. None should be prepared unless they are clearly needed. In general, the need goes up as the number of people involved in the project increases.

SECTION	CONTENTS
4.0 Systems Engineering Process	This section describes the intended execution of the systems engineering processes to be used to develop the system. These processes are generically described in the Guidebook and identified in the VEE lifecycle technical development model. The SEMP describes the processes specifically needed for a project and defines them in sufficient detail to guide the work of the systems engineering and development teams. The FHWA's Final Rule (23 CFR Part 940 part 11) places requirements on the minimum description of the systems engineering analysis for projects funded with highway trust funds. For all projects, these factors should be discussed in the SEMP. These are: ▪ Identification of portions of the regional ITS architecture being implemented or, if a regional ITS architecture does not exist, the applicable portions of the National ITS Architecture ▪ Identification of participating agencies and their roles and responsibilities ▪ Requirements definitions ▪ Analysis of alternative system configurations and technology options to meet requirements ▪ Procurement options ▪ Identification of applicable ITS standards and testing procedures ▪ Procedures and resources necessary for operations and management of the system This section will contain a description of the systems engineering procedures tailored to the specific project. There are four areas of analysis that need to be described: ▪ System Requirements Analysis – describe the methods to be used to prepare the Concept of Operations and the top-level system requirements documents. The analysis techniques that may be used include peer reviews, working groups, scenario studies, simulation and prototyping. The amount of analysis required increases with the risk of the specific requirement. The process for approving the resulting documents will be described including who is involved, whether technical reviews are necessary, and how issues and comments are resolved so the baseline can be defined. ▪ Sub-System (Functional) Analysis – describe the methods to be used to identify sub-systems and to allocate the system (top-level) requirements to the sub-systems. It is often necessary, at this step, to expand the top-level requirements into a complete description of the functions of the system, for instance, details of an operator interface. It also may be necessary at this time to define internal interfaces (sub-system to sub-system) to the same level of detail as the external interfaces (interfaces to other systems). The SEMP should describe the methods for analysis and the tools required. Budget and schedule constraints, as well as completion criteria, should be included. ▪ Design Synthesis – describe the methods to be used by the development teams to translate the functional requirements into a hardware and software design. A number of tools and methodologies exist for this and the specific ones to be used by the development team should be identified, along with the necessary resources. Describe the products to be produced as this process unfolds and the design review steps to be taken.

SECTION	CONTENTS
	▪ ▪ System Analysis – describe the methods to be used for any required technical trade-off studies, cost/benefit decisions, and risk mitigation alternative analysis. The methodologies to be used should provide a rigorous basis for selecting an alternative, a quantifiable basis for comparing the technical, cost and schedule impacts of each alternative, and comprehensive description of the risks involved with each alternative.
5.0 Transitioning Critical Technologies	This section will describe the methods and processes to be used to identify, evaluate, select and incorporate critical technologies into the system design. Since this may represent an area of considerable impact to the project, this is one of the major efforts of risk management. The need for a critical technology may be based on a performance objective. It may also be based on other factors, such as the desire to reduce acquisition or maintenance costs, the need to introduce standard compliance, or the need to meet an operational objective. In some cases, the need may be to move away from a technology that is obsolete and no longer supported by industry. Identification of candidate technologies hinges on a broad knowledge of the technologies and knowledge of each technology's status and maturity, in other words, a thorough understanding of the pros and cons of each available technology. Obtaining the resource capable of performing this step is one of the major risks encountered by project management. Sufficient analysis of the risks and benefits of a particular technology may become a major effort involving acquiring the technologies, modifying the technology to meet system requirements, and developing methods to test and evaluate the various technologies that need to be considered. Each of these steps can introduce considerable risk. Finally, incorporation of a technology into an operational system may be found to involve considerable work, especially in terms of establishing the support and maintenance environment for the technology. All of these aspects of technology introduction, especially introduction of novel technology, need to be carefully and fully addressed in the SEMP.
6.0 Integration of the System	This section describes the methods to be used to integrate the developed components into a functional system that meets the system requirements and that is operationally supportable. The systems engineering process steps to be detailed here include integration, verification, deployment and the training necessary to support operations and maintenance. Plans for validation of the system should also be covered. For each step, the resources (tools and personnel) are identified and products and criteria for each step defined.
7.0 Integration of the Systems Engineering Effort	This section addresses the integration of the multi-disciplinary organizations or teams that will be performing the systems engineering activities. Obviously, the larger the number of such organizational teams, the more important the integration of their efforts is. Each team will have both primary and support tasks from the WBS. Each team will have to be aware of the activities of other teams, especially those activities that immediately precede or follow their own primary tasks. Representative of most teams will have to be involved in critical technical reviews and in the review of baseline documentation. Likewise, up-front teams (e.g. requirements and design) must be available to support the ending activities, such as, integration, verification, deployment and training.

SECTION	CONTENTS
8.0 Applicable Documents	This section lists the applicable documents that are inputs to the project (that is, needed but not produced by the project). Such documents may include, the regional ITS architecture description, planning documents describing the project, agency procedures to be followed, standards, and specifications and other descriptions of interfacing external systems. Other applicable documents may be required by a specific project.

7.5.3 Configuration Management (CM) Plan Guidelines

Purpose of this Document

A Configuration Management Plan is one of the more common technical and management plans needed to supplement the Project Plan and the Systems Engineering Management Plan. Preferably, the agency has an established CM and CM policies in place. If that is the case, then the agency's CM Plan only needs to be supplemented with project specific information, such as organization, products and schedules. If an agency plan does not exist, then a project specific plan is developed that is focused on managing the specific configuration management issues facing this project.

Configuration management is as much of a concern after the project's system is deployed (because of maintenance and upgrades) as it is during development. If possible, the CM Plan should be written to handle both phases, development and operations.

Additional information on Configuration Management is found in section 4.8.6 of this Guidebook.

Tailoring this Document to your Project

The major challenge in writing a useable CM Plan is to create a CM process that is commensurate with the size and scope of the project. Configuration Management can become very labor intensive and expensive. Too often, the expense of CM overrides the value of CM, and it falls by the wayside. This problem is especially prominent in Change Control Management where the process is made so complex and difficult that it stifles the willingness of the developers to participate in it. Too many levels of change approval or too large a group that must approve a change are common problems. Change approval should be focused on finding workable solutions and not insisting on the perfect solution every time.

The CM processes obviously become more complex when the project involves development of custom software. However, maintaining the configuration of the Concept of Operations and the Requirements specification is applicable to almost any project.

Checklist of Critical Information

- ☑ Does the agency have an existing Configuration Management Plan that must be used by the project?
- ☑ Does the Organization section of the CM Plan identify and describe the roles of all necessary participants, including stakeholders from outside the project staff?
- ☑ Have all named participants been notified of their role and do they and their organization understand and accept this responsibility?
- ☑ Does the Configuration Item Identification section specifically name each item (documents and hardware / software products) that will be placed under configuration control, or, alternatively identify the types of items to be placed under configuration control?
- ☑ Does the Configuration Item Identification section describe when each item is placed under configuration control (baselined) and does it define the process steps that must occur before this happens?
- ☑ Does the Change Management section describe the process for preparing and submitting a proposed change request?
- ☑ Does the Configuration Status Accounting section describe the establishment of a configuration repository where the current versions of all items are kept and from which they are made available to project personnel and other stakeholders?

CONFIGURATION MANAGEMENT PLAN GUIDELINES

EIA 649 National Consensus Standard on Configuration Management

SECTION	CONTENTS
Title Page	The title page should follow the Transportation Agency procedures or style guide. As a minimum, it should contain the following information: ▪ CONFIGURATION MANAGEMENT PLAN FOR THE Name of Project AND Transportation Agency ▪ Contract number ▪ Date the document was formally approved ▪ The organization responsible for preparing the document ▪ Internal document control number, if available ▪ Revision version and date issued
1.0 Purpose of Document	A brief statement of the purpose of this document, that is, defining the processes for establishing and maintaining configuration control of the products and documentation of the project. These processes are meant to remain in place for the life of these products and documents, that is, through development, operations and upgrades.
2.0 Scope of Project	This section gives a brief description of the planned project and the purpose of the system to be built. This section may be lifted from earlier documents. It is important only to people (stakeholders) who will be introduced to the project for the first time by this document.
3.0 Organization, Roles and Responsibili-ties	This section identifies the organizational structure needed to manage and perform configuration management for this project. If possible, the members of the configuration management organization are identified by name. The section then defines the role of each member of the organization. Typically, the organization includes: ▪ A CM manager who supervises all CM activities ▪ A CM staff, reporting to the manager and who are responsible for the performance of the CM processes ▪ A change management board who, after a configuration item is baselined, approve or reject all proposed changes to that item This section also may identify any configuration management tools to be used by the project to support the CM processes, such as, a software configuration management tool.
4.0 Configuration Item Identification	This section defines the process to identify those items (outputs of the tasks of the project) that will be placed under configuration management. It also identifies when those items are to be baselined and placed under CM control. Such items include documents as well as hardware and software products. The process for placing an item under CM control are general in nature, but the specifics of the process for each item produced by this project are defined in the plan. For instance, the process for placing the project's High Level Design specification may involve review of the completed document by an identified set of stakeholders, an in-depth design review by those same stakeholders, and resolution and incorporation of all stakeholder comments. The review not only makes sure that all requirements are traced into the design but that appropriate and sufficient trade-off studies were completed concerning alternate designs. In other words, only when the stakeholders are satisfied with a particular CM item is that item declared a baseline, placed under change management control and approved for use in subsequent steps

SECTION	CONTENTS
	in the development of the system.
5.0 Change Management	This section defines the formalized process for making a change to any baselined CM item. This process generally involves generation of a change request, an in-depth analysis of the impacts of the proposed change and then formal approval (or rejection) by the change management board. The plan defines how proposed changes are to be documented, how they are submitted to the CM manager's staff, how the staff prepares them for preliminary review by the change management board, how and when the board conducts this preliminary review and how the need (as determined by the board) for further analysis is recorded, how and when this analysis is presented to the board, and how the disposition of the change request is documented and distributed by the staff.
6.0 Configuration Status Accounting	This section describes the steps to be taken by the CM manager and staff that will keep the other participants in the project aware of the configuration of the various outputs and products of the project. They will follow these defined processes to make the current configuration of documents and products known, and available, in a timely manner and to make the status of any proposed changes known as the changes are being considered by the change management board. Today, for both documents and for software products, this means having procedures for keeping, and making available, electronic files that contain the currently approved version of the item and making those files available to other project participants.
7.0 Configuration Audits	This section defines the process, and the application of that process, for verifying the configuration of a hardware or software product. This process will be invoked during verification to ensure that the product version being verified is known and is accurately described by its documentation. The processes describe how and by whom this audit is to be conducted.
8.0 Applicable Documents	This section lists other documents that are referenced in this Configuration Management Plan.

7.5.4 Needs Assessment Guidelines

Purpose of this Document

The Needs Assessment Document is a record of the stakeholder needs that motivate the development of the system. It is essential that these needs be well understood and agreed upon before system development begins. Corrections are far easier and less costly to make at this preliminary stage than they are during development or deployment.

Often the needs are vague, ill formed, or unstated. Two stakeholders who are saying the same things may actually want something entirely different. The needs assessment process clarifies these needs, and so this document is a record of what the stakeholders are actually looking for, in a clear and complete manner.

Generally it is not possible to meet all of the needs within the time and budget available for the project. Furthermore, often the stakeholders may have conflicting needs. For example, transit or emergency response want signal preemption, which conflicts with traffic management's need for smooth flow. This means that tradeoffs and prioritizations may need to be made to balance the needs that will be the focus of the system. The Needs Assessment Document will be a record of the process for selecting these key needs.

There are several purposes for the Needs Assessment Document.

- Get and document stakeholder agreement on the needs that the system is to meet, to ensure that the development starts off in the right direction, and to avoid later redirection.
- Clearly describe the needs that the system will meet, as the first step toward defining system requirements.
- Document the process and results of stakeholder consensus, relative to conflicting needs.
- Demonstrate to the stakeholders that their individual views have been incorporated.

Tailoring this Document to Your Project

Some systems are defined for a very specific and clear purpose and stakeholder, and the budget and schedule are adequate to meet that purpose. In that case, a short and simple Needs Assessment Document is sufficient, about a page or less. This will clearly state what the needs are and include an acknowledgement from the stakeholders that they concur.

More often, some sort of elicitation process is necessary to draw out the needs. There are many ways to do this (see 4.8.2), each with its own output that will be included in this document to record the process and its results. Similarly, prioritization and gap analysis results are included in this document, as a justification for the key needs selected.

In general, the form and complexity of this document reflect the amount of elicitation and analysis that was undertaken to come up with the key needs.

Checklist of Critical Information

- ☑ Are all the stakeholders identified?
- ☑ Are the needs of each stakeholder clearly described?
- ☑ Are the process and results of each elicitation activity described?
- ☑ Have essential needs (those that must be in the system) been distinguished from secondary needs (wants)?
- ☑ Is the prioritization of the secondary needs, if done, fully and clearly justified and its process documented?
- ☑ Are the process and results of the gap analysis, if done, fully described?
- ☑ Is there documentation to show that the stakeholders validated and concur with the identified key needs?
- ☑ Are the key needs, constraints, and corresponding measures of effectiveness clearly and unambiguously described?

NEEDS ASSESSMENT GUIDELINES

SECTION	CONTENTS
Title Page	The title page should follow the Transportation Agency procedures or style guide. As a minimum, it should contain the following information: ▪ NEEDS ASSESSMENT FOR THE Name of Project AND Transportation Agency ▪ Contract number ▪ Date the document was formally approved ▪ The organization responsible for preparing the document ▪ Internal document control number, if available ▪ Revision version and date issued
1.0 Purpose of Document	This section is a brief statement of the purpose of this document, that is, a description and rationale of the needs that the system will be built to meet. This is a vehicle for stakeholder feedback, and a justification for the key needs selected.
2.0 Overview	This section gives a brief overview of the system to be built, and describes the stakeholders and the expected role of each.
3.0 Referenced Documents	This optional section is a place to list any supporting documentation used and other resources that are thought useful in understanding the operations of the system.
4.0 Needs elicitation	This is the description and discussion of all elicitation activities. The process used and the results are included, possibly backed up by specifics (e.g., records of interviews) included in the appendix.
5.0 Needs description	This section is the heart of the document. It describes clearly and fully the needs expressed by the stakeholders, as they stand after the elicitation and validation processes. The essential needs are highlighted, and distinguished from those that are "wants" that may be sacrificed for cost or for more critical needs. This section also includes all system constraints that are known at this point. In addition, as much as possible, the needs should have corresponding measures of effectiveness or measures of performance that provide metrics for how well, or whether, the need is met.
6.0 Needs validation	This section describes the process and results of validating the collected needs with the stakeholders. Any changes that came out of this process should be incorporated into 5.0.
7.0 Gap analysis	This optional section describes the current system, compares it with the needs, and identifies the most pressing gaps to fill, in terms of criticality of the need and the extent of the gap. This section is not needed if the needs in 5.0 are consistent with each other and with budget and schedule.
8.0 Cost comparison	This optional section may be used if there are conflicting needs. This gives a rough order of magnitude lifecycle cost estimate for each option. Alternatively, ease of implementation, or some other stand-in for cost, may be used. This section may also be used to document any analysis that was done to verify that the identified needs can be met within the budget.
9.0 Selection of key needs	This optional section is used if the needs must be prioritized. This refers back to 7.0 and 8.0 and documents the process and results of prioritizing the needs, and the rationale for the selection. It describes the selected key needs.
10.0 Validation of	This optional section documents the final feedback of the stakeholders relative to the key needs described in 9.0. This is used if the needs must be

SECTION	CONTENTS
key needs	prioritized. This section documents the stakeholders' agreement that the system will focus on the identified key needs.
11.0 Appendix	The appendix is optional. This is a good place to put back-up material from the elicitation and analyses, so that the main sections are succinct, while full justification of the results is available here for those interested. It also may include a glossary or notes, if appropriate.

7.5.5 Concept of Operations Guidelines

Purpose of this Document

The Concept of Operations is a description of how the system will be used. It is non-technical, and presented from the viewpoints of the various stakeholders. This provides a bridge between the often vague needs that motivated the project to begin with and the specific technical requirements. There are several reasons for developing a Concept of Operations.

- Get stakeholder agreement on how the system is to be operated, who is responsible for what, and what the lines of communication are.
- Define the high-level system concept and justify that it is superior to the other alternatives.
- Define the environment in which the system will operate.
- Derive high-level requirements, especially user requirements.
- Provide the criteria to be used for validation of the completed system.

Tailoring this Document to Your Project

The greater the expected impact on operations, the more detailed the Concept of Operations needs to be. For example, automating operations that were formerly manual or integrating activities that were formerly independent will require the involvement of the various operators, clear and detailed description of their new procedures, and possibly examination of alternative approaches. This is especially true when building a regional system by integrating existing local systems. Local operations will usually change after integration, for compatibility and to take advantage of newly available regional resources.

For a simple system that requires little operator involvement and no coordination, this document may only be a couple of pages long. The key is to describe all possible system operations, both normal and failure, as seen by each stakeholder.

Checklist of Critical Information

- ☑ Is the reason for developing the system clearly stated?
- ☑ Are all the stakeholders identified and their anticipated roles described? This should include anyone who will operate, maintain, build, manage, use, or otherwise be affected by the system.
- ☑ Are alternative operational approaches (such as centralized vs. distributed) described and the selected approach justified?
- ☑ Is the external environment described, including required interfaces to existing systems?
- ☑ Is the support environment described, including maintenance?
- ☑ Is the operational environment described?
- ☑ Are there clear and complete descriptions of normal operational scenarios?
- ☑ Are there clear and complete descriptions of maintenance and failure scenarios?
- ☑ Do the scenarios include the viewpoints of all involved stakeholders, and make it clear who is doing what?
- ☑ Are all constraints on the system development identified?

CONCEPT OF OPERATIONS GUIDELINES

Relevant standards are the ANSI/AIAA G-043-1992 standard and IEEE Standard P1362 V3.2.

SECTION	CONTENTS
Title Page	The title page should follow the Transportation Agency procedures or style guide. As a minimum, it should contain the following information: ▪ CONCEPT OF OPERATIONS FOR THE Name of Project AND Transportation Agency ▪ Contract number ▪ Date the document was formally approved ▪ The organization responsible for preparing the document ▪ Internal document control number, if available ▪ Revision version and date issued
1.0 Purpose of Document	This section is a brief statement of the purpose of this document, that is, a description and rationale of the expected operations of the system under development. It is a vehicle for stakeholder discussion and consensus to ensure that the system that is built is operationally feasible. This will briefly describe contents, intention, and audience. One or two paragraphs will suffice.
2.0 Scope of Project	This short section gives a brief overview of the system to be built, including its purpose and a high-level description. It describes what area will be covered and which agencies will be involved, either directly or through interfaces. One or two paragraphs will suffice.
3.0 Referenced Documents	This optional section is a place to list any supporting documentation used and other resources that are thought useful in understanding the operations of the system. This could include any documentation of current operations and any strategic plans that drive the goals of the system under development.
4.0 Background	Here is a brief description of the current system or situation, how it is used currently, and its drawbacks and limitations. This leads into the reasons for the proposed development, and the general approach to improving the system. This is followed by a discussion of the nature of the planned changes and a justification for them.
5.0 Concept for the proposed system	This section describes the concept exploration. It starts with a list and description of the alternative concepts examined. The evaluation and assessment of each alternative follows. This leads into the justification for the selected approach. The operational concept for that selected approach is described here. This is not a design, but a high-level, conceptual, operational description, in only as much detail as needed to be able to develop meaningful scenarios. In particular, if alternative approaches differ in terms of which agency does what, that will need to be resolved and described. An example would be the question of whether or not a regional signal system will have centralized control.
6.0 User-Oriented Operational Description	This section focuses on how the goals and objectives are accomplished currently. Specifically, it describes strategies, tactics, policies, and constraints. This is where the stakeholders are described. It includes who users are and what the users do, specifically when and in what order operations take place, personnel capabilities, organizational structures, personnel and inter-agency interactions, and types of activities. This may also include operational process models in terms of sequence and interrelationships.

SECTION	CONTENTS
7.0 Operational Needs	Here is a description of the vision, goals and objectives, and personnel needs that drive the requirements for the system. Specifically, this describes what the system needs to do that it does not do now.
8.0 System Overview	This is an overview of the system to be developed. This describes its scope, who the users are, what it interfaces with, its states and modes, the planned capabilities, its goals and objectives, and the system architecture. Note that the system architecture is not a design, which will be done later, but provides a structure for describing the operations, in terms of where the operations will be carried out, and what the lines of communication will be.
9.0 Operational Environment	This section describes the physical operational environment in terms of facilities, equipment, computing hardware, software, personnel, operational procedures and support necessary to operate the deployed system. For example, it will describe the personnel in terms of their expected experience, skills, and training, typical work hours, and other activities (e.g., driving) that must be or may be performed concurrently.
10.0 Support Environment	This describes the current and planned physical support environment. This includes facilities, utilities, equipment, computing hardware, software, personnel, operational procedures, maintenance and disposal. This includes expected support from outside agencies.
11.0 Operational Scenarios	This is the heart of the document. Each scenario describes a sequence of events, activities carried out by the user, the system, and the environment. It specifies what triggers the sequence, who or what performs each step, when communications occur and to whom or what (e.g., a log file) and what information is being communicated. The scenarios will need to cover all normal conditions, stress conditions, failure events, maintenance, and anomalies and exceptions. There are many ways for presenting scenarios, but the important thing is that each stakeholder can clearly see what his expected role is to be.
12.0 Summary of impacts	This is an analysis of the proposed system and the impacts on each of the stakeholders. It is presented from the viewpoint of each, so that they can readily understand and validate how the proposed system will impact their operations. Here is where any constraints on system development are documented. Metrics for assessing system performance are also included here.
13.0 Appendices	This is a place to put a glossary, any notes, and backup or background material for any of the sections. For example, if might include analysis results in support of the concept exploration.

7.5.6 Requirements Guidelines

Purpose of This Document

This document describes what the system is to do (functional requirements), how well it is to perform (performance requirements) and under what conditions (non-functional and performance requirements). This document does not define how the system is to be built. This document pulls together requirements from a number of sources including but not limited to:

- Concept of Operations and Scenarios
- Elicitation process – previous studies, “Day in the Life” studies, interviews, and workshops.
- Constraints that are put onto a project, such as policies that will drive constraints on the system. (Example – The Agency policy is to use Oracle in Intelligent Transportation Systems)

Intelligent Transportation System projects have a Requirements Specification at the system and sub-system levels.

This document sets the technical scope of the system to be built and is the basis for verifying the system and sub-systems when delivered (via the Verification Plan).

Tailoring this Document to Your Project

Any ITS projects will need a set of requirements defining what is needed. The tailoring is in how extensive to document these requirements. One way to gauge how many requirements to write and/or how much detail to have in the requirements document is to start at the finish line. That is, starting at the top level system or system of systems, ask yourself, what are all the functions that you would like to have done in order to satisfy the agency that the system is doing what you expect it to do, how well it needs to do these functions and under what conditions. Each of these tests will need a requirement. This is done for the system and the sub-systems. For simple systems there may only be 1 or 2 pages of requirements that can fully define what the system is to do. In more complex systems this could be 10 to 20 pages or more.

Other factors that drive the extent to which requirements need to be written are the amount of COTS products that are used. Since these off-the-shelf products have their own specifications, it may be sufficient to reference them after they have been reviewed to the point where it has been determined that the product will meet the intended need of the agency. For example, the traffic control systems that are on the market have sufficient documentation to cover the majority of functions that are required. The additional requirements would be for any modifications or enhancements needed.

Checklist of Critical Information

- ☑ Is there a definition of all the major system functions?
- ☑ With each function of the system is there a set of requirements that describes what the function does, who well it is to do it and under what conditions (e.g. environmental, reliability, and availability.)
- ☑ Are all terms, definitions and acronyms defined?
- ☑ Are all supporting documents such as standards, concept of operations, and others referenced?
- ☑ Does each requirement have a link (traceability) to a higher level requirement of a user-specified need?
- ☑ Is each requirement concise, verifiable, clear, feasible, necessary, unambiguous, and technology independent?
- ☑ Are all technology dependent requirements identified as constraints?
- ☑ Does each requirement have a method of verification defined?

SYSTEM AND SUB-SYSTEM REQUIREMENTS GUIDELINES

IEEE Std 1233 Guide for developing System Requirements Specifications

SECTION	CONTENTS
Title Page	The title page should follow the Transportation Agency procedures or style guide. As a minimum, it should contain the following information: ▪ SYSTEM REQUIREMENTS/SUB-SYSTEM REQUIREMENTS Name of Project AND Transportation Agency ▪ Contract number ▪ Date the document was formally approved ▪ The organization responsible for preparing the document ▪ Internal document control number, if available ▪ Revision version and date issued
1.0 Scope of System or Sub-system	▪ Contains a full identification of the system ▪ Provides a system overview and briefly states the purpose of the system ▪ Describes the general nature of the system ▪ Summarizes the history of system development, operation, and maintenance ▪ Identifies the project stakeholders, acquirer, users, support agencies ▪ Identifies current and planned operating sites
2.0 Reference	Identifies all needed standards, policies, laws, concept of operations, concept exploration documents and other reference material that supports the requirements.
3.0 Requirements	Functional requirements (What the system shall do) Performance requirements (How well the requirements should perform) Interface requirements (Definition of the interfaces) Data requirements (Data elements and definitions of the system) Non-Functional requirements – reliability, safety, environmental (temperature) Enabling requirements (Production, development, testing, training, support, deployment, disposal). This can be done through references to other documents or embedded in this requirements Constraints – (e.g. Technology, design, tools, and/or standards)
4.0 Verification methods	For each requirement, identify one of the following methods of verification: **Demonstration** – a requirement that the system can demonstrate without external test equipment. **Test** – a requirement that requires some external piece of test equipment. E.g logic analyzer, and/or volt meter. **Analyze** – a requirement that is met indirectly through a logical conclusion or mathematical analysis of a result. E.g. Algorithms for congestion: the designer may need to show that the requirement is met through the analysis of count and occupancy calculations in software of firmware. **Inspection** – verification through a visual comparison. For example, quality of welding may be done through a visual comparison against an in-house standard.
5.0 Supporting Documentation	Catch-all for anything that may add to the understanding of the Requirements without going elsewhere (Reference section)

SECTION	CONTENTS
	Examples: diagrams, analysis, key notes, memos, rationale, stakeholders contact list
6.0 Glossary	Terms, acronyms, definitions

7.5.7 Design Specifications Guidelines

Purpose of these Documents

These documents describe how the system is to be built. They take the requirements (what the system will do) and translate them into a hardware and software design that can be built. Collectively, the purpose of these documents is to:

Provide a documented description of the design of the system that can be reviewed and approved by the stakeholders

- Provide a description of the system in enough detail that its component parts can be procured, and built
- Provide a description of the hardware and software system components in sufficient detail for them to be maintained and upgraded.
- For most projects, two levels of design specification are developed. The High Level Design Specification Document supports the first purpose and the Detailed Design Specification Document supports the latter two.

For some systems, it is advisable to create separate documents, called Interface Design Documents, to describe the internal and external interfaces of the system being built.

Tailoring these Documents to Your Project

Any ITS projects that are structured to produce a physical hardware / software system require some level of design description of the system to be procured or built. Study projects with only paper products don't need them. For simple systems or for systems that are completely off-the-shelf, only one minimal document is sufficient (perhaps just a list of the items to be procured), but even that is still required.

If a project involves fabrication of hardware components, then the information contained in the design specifications are supplemented with drawings from which the parts are built. Construction and installation drawing may also be required.

However, if a project involves development of custom software, even relatively simple software, then both documents are strongly recommended.

A software design is documented by these specifications and by the source code itself. It is vital that the Detailed Design specification exists along with the source code. Further, the specification must track to this code.

Interface (both internal and external) descriptions may be completely contained in the Detailed Design specifications or they may be described in a separate document, the Interface Design Document. Some modern programming techniques make processor-to-processor interfaces completely transparent to the code. However, some interface methods, especially interfaces to existing external systems, are very specialized and unique. In these cases, a separate document that can be easily reviewed by engineers on both sides of the interface is very useful.

Checklist of Critical Information

- ☑ Does the High Level Design Document include definition of requirements unique to the chosen architecture (interfaces between sub-systems, for instance)?
- ☑ Is the definition of each requirement from the Requirements Document complete enough for implementation or does it need to be expanded in the High Level Design Document?
- ☑ Are all the system requirements traced to the sub-systems in the High Level Design Document?
- ☑ Are all necessary COTS software products identified in the High Level or the Detailed Design Document?
- ☑ Is the design approach for common software methods defined, as appropriate, in both the High Level and Detailed Design Documents?
- ☑ Is the architecture, both hardware and software, of the sub-systems (components and interconnections) defined in the high level design specification?
- ☑ Are any necessary database schema and structures defined in the High Level and Detailed Design Documents?
- ☑ Are the hardware components defined in enough detail in the design documents to support procurement or fabrication?
- ☑ Has the trace from requirements to hardware and software components been checked and verified?
- ☑ Is the Detailed Design Document linked to the source code components, that is, do they use the same object names, file names, attribute names, and method names?

HIGH LEVEL DESIGN SPECIFICATION GUIDELINES

IEEE Std 1233 Guide for developing System Requirements
IEEE 1471-2000 Recommended Practice for Architectural Description of Software Intensive Systems

SECTION	CONTENTS
Title Page	The title page should follow the Transportation Agency procedures or style guide. As a minimum, it should contain the following information: ▪ HIGH LEVEL DESIGN SPECIFICATION FOR THE Name of Project AND Transportation Agency ▪ Contract number ▪ Date the document was formally approved ▪ The organization responsible for preparing the document ▪ Internal document control number, if available ▪ Revision version and date issued
1.0 Purpose of Document	This section is a brief statement of the purpose of this document, that is, a high level description of the architecture (hardware and software) of the system. It also summarize the contents of the document. Sometimes the High Level Design specification is used to document some requirements not covered elsewhere, such as an operator interface or interfaces to external systems. It also may be necessary to include functional requirements arising from the internal interfaces created between the sub-systems.
2.0 Scope of Project	This section gives a brief description of the planned project and the purpose of the system to be built. This section can be copied from a previous document, and is included for completeness as this may be the only document which some project participants and stakeholders may see.
3.0 Sub-Systems	This section describes the architecture of the system and how it is divided into sub-systems, if that is found to be necessary. Simpler systems may not need to be subdivided, and if so, this section is void. If sub-systems are needed, then each is described in terms of its purpose, its functionality, its interfaces with other sub-systems, and its component parts (hardware and software). If the requirements call for different capabilities at multiple sites, then the allocation of the sub-systems to these sites is shown. In order to describe the functionality of a sub-system, it is necessary to allocate system requirements to each sub-system. All requirements must be covered by at least one sub-system, however, some requirements (and especially performance requirements) may be applicable to several subsystems. An explicit trace of all requirements from the Requirements Document into the sub-systems is a part of this document. In addition to the system requirements, additional requirements may be necessary to show how the sub-systems work together. Those types of requirements are analyzed and documented here.
4.0 Hardware Components	This section identifies the hardware components of each sub-system. It identifies them by name, by function, by capabilities, by source (manufacturer), by quantity and shows the interconnections between the components (e.g. point-to-point, or local area network). If a hardware component needs optional components or features, they are listed and defined at this time. This section also includes a trace of requirements, where applicable, into the hardware components.
5.0 Software	This section describes the preliminary design of the software application. It

SECTION	CONTENTS
Components	shows the allocation of the software to sub-systems and to hardware elements. It shows and identifies the commercial off-the-shelf (COTS) software packages to be used and shows their allocation to sub-systems and to hardware components. It also shows and identifies all custom designed software packages and shows their allocation to sub-systems and to hardware components. It shows the architectural relationship between the various software packages, both custom and COTS. The high level design of each custom software package is described. The method used for this description depends on the methodology being used for software design. That methodology may be object oriented design, data flow design, structured design, or any other method chosen by the project and the software development team. For example, if an object oriented software design methodology is to be used, the description of the custom software components for the High Level Design specification would include: Preliminary class description for significant internal and external classes necessary to implement the functional requirements ▪ Preliminary description of the attributes, methods and relationships of each class of objects ▪ Class diagrams and other diagramming methods as appropriate, such as, sequence, package, activity concurrency and state diagrams ▪ Component diagrams to describe the physical partitioning of the software into code components ▪ Descriptions of common patterns to be used in the software design, such as, the pattern to be used for inter-process communication, or for implementation of an operator interface. ▪ Trace requirements into each software package
6.0 Sub-System Requirements	This document may be used to describe additional requirements that were not covered in the requirements specifications. These may include, but are not limited to: ▪ Showing greater detail of previously defined functional requirements based on additional functional analysis, for instance, defining the details of a complex algorithm ▪ Providing complete details of complex requirements, such as a detailed description of a complex operator interface where considerable work with operations personnel is necessary before a definitive statement of the requirement can be made ▪ Providing complete details of an interface with an external system ▪ Stating requirements which result from the separation of the system into sub-systems, that is, identifying functional requirements for the way these sub-systems work together Of course, these types of requirements (with the exception of the last type) also may be included in the Requirements Document or may be documented in separate documents, as deemed appropriate.
7.0 Applicable Documents	This section lists the applicable documents that constrain the design process. Such documents may include standards and external system specifications.

DETAILED DESIGN SPECIFICATION GUIDELINES

SECTION	CONTENTS
Title Page	The title page should follow the Transportation Agency procedures or style guide. As a minimum, it should contain the following information: ▪ DETAILED DESIGN SPECIFICATION FOR THE Name of Project AND Transportation Agency ▪ Contract number ▪ Date the document was formally approved ▪ The organization responsible for preparing the document ▪ Internal document control number, if available ▪ Revision version and date issued
1.0 Purpose of Document	This section is a brief statement of the purpose of this document, that is, to expand and complete the preliminary design descriptions included in the High Level Design Document.
2.0 Scope of Project	This section describes the project and may be lifted from the High Level Design Document.
3.0 Sub-Systems	This section completes the description of the system architecture and of the sub-systems, as necessary.
4.0 Hardware Components	This section completes the description of the hardware components. It contains a detailed list of the exact hardware items to be procured by name, part number, manufacturer and quantity. If necessary, it lists any hardware component specifications or drawings which have been prepared by the design team.
5.0 Software Components	This section completes the description of the software components. It contains a detailed list of the COTS software products to be procured, by vendor, name, part number and options. If the project involves custom software applications this section becomes the dominant and largest part of the Detailed Design Document. Its purpose is to provide enough information so the code can be developed and, subsequently, so the code can be understood for maintenance and for system upgrades. As a result, the overriding requirement is that the descriptions of the software components are complete and the link between these descriptions and the actual source code is clear and explicit. The Detailed Design specification is primarily a completion of the preliminary information in the High Level Design specification. Any corrections to the information in the previous document should be made at this time. Again, if a software design tool is used, it may produce most of the Detailed Design specification. For example, if an object oriented software design methodology is to be used, the description of the custom software components for the Detailed Design specification would include expansion of the following from the High Level Design specification: ▪ Class description for significant internal and external classes necessary to implement the functional requirements ▪ Description of each class' attributes, methods and relationships ▪ Class diagrams and other diagramming methods as appropriate, such as, sequence, package, activity concurrency and state diagrams ▪ Component diagrams to describe the physical partitioning of the software

SECTION	CONTENTS
	into code components ▪ Descriptions of common patterns to be used in the software design, such as, the pattern to be used for inter-process communication, or for implementation of an operator interface.

INTERFACE DESIGN DOCUMENT GUIDELINES

SECTION	CONTENTS
Title Page	The title page should follow the Transportation Agency procedures or style guide. As a minimum, it should contain the following information: ▪ INTERFACE DESIGN DOCUMENT FOR THE Name of Interface FOR THE Name of Project AND Transportation Agency ▪ Contract number ▪ Date the document was formally approved ▪ The organization responsible for preparing the document ▪ Internal document control number, if available ▪ Revision version and date issued
1.0 Purpose of Document	This section is a brief statement of the purpose of this document, that is, to define the function and design of an interface between two parts of the system or between the system and an external system.
2.0 Scope of Project	This section describes the project and may be lifted from the High Level Design Document.
3.0 Interface Purpose and General Description	This section is used to describe, in operational terms, the purpose of this interface and how that purpose relates to the overall operation of the system being designed. It describes the information flow, in both directions if that is applicable, and the actions or conditions that cause information to be transferred across the interface. It describes where that information comes from and where it is used.
4.0 Communications Method	This section describes the communications protocols associated with information flow across the interface, especially protocols that the programmer has to use in order to make the transfer occur. This form and content of this section, and the next, are very dependent on the type of communication method used. For instance, the description of a database replication method is different from a File Transfer Protocol (FTP) method, or from a remote procedure call method. There are many other communications methods that can be used. For internal interfaces, selection of process-to-process communications method is part of the software design effort. However, when communicating with an external system, the usual case is that system already exists and has a defined communication protocol. In this case, the software designer must build a compatible interface. That work is facilitated by this document.
5.0 Specific Interface Design	Along with the previous section, the form and content of this is completely dependent on the method used to transfer information, or data, from process to process and from system to system. This section focuses on the form and content of the data elements themselves instead of the communications protocols described before. For instance, if database replication is used, this section describes the logical data structure and the specific database information contained in the fields of the database. If a message method is used, this section describes the content of each field of the message and its allowable values. If a remote procedure call type of interface is implemented, this sections describes the function of the call, the parameters passed with it, the parameters returned by the call and the actions taken by the remote procedure. These are just three examples of a great variety of methods that may be used. This section must contain enough information to allow the software developer to design and write code to implement the interface.

7.5.8 Integration Plan Guidelines

Purpose of this Document

A project's integration and verification strategy is very closely tied to the design of the system and its decomposition into sub-systems. The factors that are considered when developing the sub-system design are covered elsewhere in this Guidebook. Whatever the goals were (and they vary from project to project), the Integration Plan needs to be structured to bring the components together to create each sub-system and to bring the various sub-systems together to make the whole system. Further, this needs to be done in a way that supports the deployment strategy.

That is the first purpose of an Integration Plan. The second purpose is to describe to the participants in each integration step what has to be done. The integration team has to assemble various resources for each integration step and the Integration Plan lays out what these resources are and when, and where, they will be needed.

Tailoring this Document to Your Project

An Integration Plan, at least as a separate written document, is not always needed. The complexity of the system, the complexity of the eventual deployment of the system, and even the complexity of the development effort influence the decision to prepare an Integration Plan. For instance, a deployment strategy that calls for multiple installations at multiple locations can require a rather complex sequence of integration activities. Another common complexity of integration arises when different teams are developing the sub-systems. This is especially true when the different development teams are different contractors, each with their own contract. In this case, they need to know more about their required work to support integration than would be the case if the same development team is working both sides of the integration effort. The same type of complexity comes into play when an integration step involves external systems owned by other agencies, or at least other organizations within your agency.

If a separate Integration Plan is not warranted, the necessary planning information can be included in the Project Plan, in the SEMP, in the Verification Plan and in the software development plans of the development team.

Checklist of Critical Information

- ☑ Does the Integration Plan include and cover integration of all of the components and sub-systems, either developed or purchased, of the project?
- ☑ Does the Integration Plan account for all external systems to be integrated with your system (for example, communications networks, field equipment, other complete systems owned by your agency or owned by other agencies)?
- ☑ Does the Integration Plan fully support the deployment strategy, in terms of providing integrated systems and sub-systems when and where they are to be deployed?
- ☑ Does the Integration Plan mesh with the Verification Plan?
- ☑ For each integration step, does the Integration Plan define what components and sub-systems are to be integrated?
- ☑ For each integration step, does the Integration Plan identify all the needed participants and define what their roles and responsibilities are?
- ☑ Does the Integration Plan establish the sequence and schedule for every integration step?
- ☑ Does the Integration Plan spell out how integration problems are to be documented and resolved?

INTEGRATION PLAN GUIDELINES

SECTION	CONTENTS
Title Page	The title page should follow the Transportation Agency procedures or style guide. As a minimum, it should contain the following information: ▪ INTEGRATION PLAN FOR THE Name of Project AND Transportation Agency ▪ Contract number ▪ Date the document was formally approved ▪ The organization responsible for preparing the document ▪ Internal document control number, if available ▪ Revision version and date issued
1.0 Purpose of Document	A brief statement of the purpose of this document, that is, the plan for integrating the components and sub-systems of the project prior to verification.
2.0 Scope of Project	This section gives a brief description of the planned project and the purpose of the system to be built. Special emphasis is placed on the project's deployment complexities and challenges. This section may be lifted from earlier documents. It is important only to people (stakeholders) who will be introduced to the project for the first time by this document.
3.0 Integration Strategy	This section informs the reader of what the high level plan is for integration and, most importantly, why the integration plan is structured as it is. As mentioned before, the Integration Plan is subject to several constraints, sometimes conflicting constraints. Also, it is one part of the larger process of build, integrate, verify and deploy, all of which must be synchronized to support the same project strategy. So, for even a moderately complex project, the integration strategy, based on a clear and concise statement of the project's goals and objectives, is here described at a high, but all-inclusive, level. It may also be necessary to describe the analysis of alternative strategies to make it clear why this particular strategy was selected. However, since the same strategy is the basis for the Build Plan, the Verification Plan and the Deployment Plan, it may only be necessary to justify this strategy once, perhaps in the Project Plan or in the SEMP. This section covers and describes each step in the integration process. It describes what components are integrated at each step and gives a general idea of what threads of the operational capabilities (requirements) are covered. It ties the plan to the previously identified goals and objectives so that the stakeholders can understand the rationale for each integration step. This summary level description also defines the schedule for all the integration efforts.
4.0 Phase 1 Integration	This, and the following sections, define and explain each step in the integration process. The intent here is to identify all the needed participants and to describe to them what they have to do. In general, the description of each integration step should identify: ▪ The location of the activities ▪ The project-developed equipment and software products to be integrated. Initially this is just a high level list but eventually the list must be exact and complete, showing part numbers and quantity. ▪ Any support equipment (special software, test hardware, software stubs

<table>
<tr><th>SECTION</th><th>CONTENTS</th></tr>
<tr><td></td><td>and drivers to simulate yet-to-be-integrated software components, external systems) needed for this integration step. The same support equipment is most likely needed for the subsequent verification step.
▪ All integration activities that need to be performed after installation, including integration with on-site systems and with external systems at other sites.
▪ A description of the verification activities (as defined in the applicable Verification Plan) that occur after this integration step.
▪ The responsible parties for each activity in the integration step
▪ The schedule for each activity</td></tr>
<tr><td>5.0 Phase N Integration</td><td>This, and any needed additional sections, follows the format for section 4. Each covers a step in the integration effort.</td></tr>
</table>

7.5.9 Verification Documents Guidelines

Purpose of these Documents

These documents plan, describe and record the activity of verifying that the system built meets the specified requirements. Since a complex system may involve a series of verification activities, several sets of these verification documents may be needed. All of these verification documents follow the master plan for verification defined in the Systems Engineering Management Plan

Usually, for even moderately complex systems, three levels of verification documents are prepared, a plan to initially lay out the specific verification effort, a procedure that is the specific and detailed steps to be followed to perform the test, and a report on the results of the testing activity. All three are described here.

A critical issue is to be sure that all requirements are verified by the testing activity. This is best done by including a trace of each requirement into a test case, and eventually, into a step in the Verification Procedure.

Additional Information is found in IEEE 1012-1998, Software Verification and Validation.

Tailoring these Documents to Your Project

For the very simplest projects, and especially where the system is essentially off-the-shelf and does not involve any custom software development, and where the project office personnel have a very clear understanding of the purpose of the system, a separate Verification Plan and procedure may not be required. In some cases, it is possible to take a copy of the Requirements Document, improvise procedures, and annotate the Requirements Document with the results of each test step. This can be a perfectly acceptable way to verify the operations of a system.

However, if the system is more complex, if there are a number of separate verification activities, if multiple deployment sites are involved, and especially if more than one or two stakeholders who have to be satisfied, then preparation of these verification documents is strongly advised.

There is also the question of how comprehensive to make the verification effort. It is impossible to test everything, that is, all possible combinations of actions under all possible operational situations. A good rule of thumb is that if it was important enough to write down as a requirement, then it should be tested at least once, as part of a reasonable operational scenario. This may not, for example, test all possible failure mode conditions, but, if a good job was done in writing the requirements, then the most important and most likely are verified.

Checklist of Critical Information

Verification Plan

- ☑ Does the Verification Plan answer all the questions of who, what, where, and when concerning test conduct?
- ☑ Does the Verification Plan make it clear what needs to happen if a test failure is encountered?
- ☑ Does the Verification Plan define the configuration of the hardware, software, and external system needed for each test case?
- ☑ Are all applicable requirements traced to a test case in the Verification Plan and does each test case define a realistic and doable test?

Verification Procedure

- ☑ Is each step in the Verification Procedure traced to a test case and to a requirement?
- ☑ Are all the necessary initial conditions and set-up defined for each procedure?
- ☑ Has each verification procedure been dry run prior to the formal test and have the procedures been updated as a result?

Verification Report

- ☑ Does the Verification Report describe in detail the resolution of every test anomaly encountered during testing?

VERIFICATION PLAN GUIDELINES

IEEE 1012-1998 Independent Verification and Validation

SECTION	CONTENTS
Title Page	The title page should follow the Transportation Agency procedures or style guide. As a minimum, it should contain the following information: ▪ VERIFICATION PLAN FOR THE Name of Project AND Transportation Agency ▪ Contract number ▪ Date the document was formally approved ▪ The organization responsible for preparing the document ▪ Internal document control number, if available ▪ Revision version and date issued
1.0 Purpose of Document	This section identifies the type of verification activity to be performed within this Verification Plan. For instance, this activity may verify the entire system, a sub-system, the deployment at a site, a burn-in test, or any other verification activity called for in the Program Plan or in the SEMP.
2.0 Scope of Project	This section gives a brief description of the planned project and the purpose of the system to be built. Special emphasis is placed on the project's complexities and challenges that must be addressed, and verified, by the systems engineering efforts. This section also describes the environment in which the project operates. It identifies the organization structures that encompass all stakeholders and gives a brief description of the role to be played by each stakeholder. This includes ad hoc and existing management work groups and multi-disciplinary technical teams that should be formed or used to support the project. Such teams are critical to reaching successful system deployment.
3.0 Referenced Documents	This is a list of all documents used in the preparation of this Verification Plan. This almost always includes the Project Plan, the SEMP (if one was written), and the applicable Requirements Documents. However, reference of other documents, such as descriptions of external systems, standards, a Concept of Operations, and manuals, may need to be included.
4.0 Test Conduct	This section provides details on how the testing is accomplished. It defines who does the testing, when and where it is to be done, the responsibilities of each participant before, during, and after each test, the hardware and software to be used (and other systems as well), and the documents to be prepared as a record of the testing activity. Another very important part of this section defines how testing anomalies are to be handled (that is, what to do when a test fails). In general, the following information should be included in this section: ▪ A description of the participating organizations and personnel and identification of their roles and responsibilities. This may include for example, a test conductor, test recorder, operators, and/or engineering support. ▪ Identification of the location of the testing effort, that is, the place, or places, where the testing progress must be observed. ▪ The hardware and software configuration for all of the test cases, including hardware and software under test and any supporting test equipment, software, or external systems. Several configurations may be necessary.

<table>
<tr><th>SECTION</th><th>CONTENTS</th></tr>
<tr><td></td><td>

Identification of the documents to be prepared to support the testing, including Verification Procedures, a Verification Report and descriptions of special test equipment and software.
Details on the actual conduct of the testing, including:

Notification of participants
Emphasis on the management role of the test conductor
Procedures for approving last minute changes to the procedures
The processes for handling a test failure, including recording of critical information, determination of whether to stop the testing, restart, or skip a procedure, resolution of the cause of a failure (e.g. fix the software, reset the system, and/or change the requirements), and determination of the retesting activities necessary as a result of the failure.

</td></tr>
<tr><td>5.0 Test Identification</td><td>
This section is the heart, and largest, section of the Verification Plan. Here we identify the specific test cases to be performed. A test case is a logical grouping of functions and performance criteria (all from the Requirements Documents) that is to be tested together. For instance, a specific test case may cover all the control capabilities to be provided for control of a changeable message sign. There may be several individual requirements that define this capability, and they all are verified in one test case. The actual grouping of requirements into a test case is arbitrary, but they should be related and easily combined into a reasonable set of test procedure actions.

Each test case should contain at least the following information:

A description name and a reference number
A complete list of the requirements to be verified. For ease of tracing of requirements, into the Verification Plan and into other documents, the requirements are given numbers so they can be accurately and conveniently referenced without repeating all the words of the requirement.
A description of the objective of the test case, usually taken from the wording of the requirements, to give the reader an understanding of the scope of the test case.
Any data to be recorded or noted during the test, such as expected results of a test step. Other data, such as a recording of a digital message sent to an external system, may be required to verify the performance of the system.
A statement of the pass/fail criteria. Often this is just a statement that the system operates per the requirements.
A description of the test configuration, that is, a list of the hardware and software items needed for the test and how they should be connected. Often, the same configuration is used for several tests.
A list of any other important assumptions and constraints necessary for conduct of the test case.

</td></tr>
</table>

VERIFICATION PROCEDURE GUIDELINES

SECTION	CONTENTS
Title Page	The title page should follow the Transportation Agency procedures or style guide. As a minimum, it should contain the following information: ▪ VERIFICATION PROCEDURE FOR THE Name of Project AND Transportation Agency ▪ Contract number ▪ Date the document was formally approved ▪ The organization responsible for preparing the document ▪ Internal document control number, if available ▪ Revision version and date issued
1.0 Purpose of Document	This section identifies the type of verification to be performed. For instance, this activity may verify the entire system, a sub-system, the deployment at a site, a burn-in test, or any other verification activity called for in the Program Plan or in the SEMP.
2.0 Verification Configuration and Software Under Test	This section identifies the equipment and software to be verified. It also identifies all equipment and software necessary for this verification activity that is external to the system / sub-system configuration under test. This may include special test equipment and any external systems with an interface to the configuration under test. For the hardware / software configuration under test, this section identifies: ▪ Each hardware item by part number and serial number ▪ Each item of commercial-off-the-shelf (COTS) software, by part number and version number ▪ Each source code file of custom developed software, by file name and version number ▪ For all special test equipment / software, this section identifies: ▪ Each hardware item by part number, version number and serial number, version number ▪ Each item of COTS software, by part number and version number ▪ Each source code file of custom developed software, by file name and version number For each external system interface, this section identifies: ▪ The name and location of the external system
3.0 Verification Setup	This section describes the steps to be taken to set up each verification configuration, including, but not limited to, tuning of the hardware, configuring and starting the software, starting the special test software and set-up steps at each external system to be used.
4.0 Verification Procedures	This section describes the step-by-step actions to be taken by the verification operator for each verification case. Each step includes: ▪ Operator action to be taken. This operator action may be, for example, an entry at a workstation, initiation of a routine in the special test software, or an action at an external system. ▪ Expected result to be observed. This too may take several forms, for example, display of certain information at a workstation, a response at an external system, recording of data for subsequent analysis, or an action by a field device. ▪ Pass / fail entry space. Here the verification conductor records whether or not the expected result occurred. If the expected results are not observed,

SECTION	CONTENTS
	then the procedures for dealing with failures contained in the Verification Plan are invoked. ▪ A trace of each verification step from a verification case in the applicable Verification Plan and a trace from a requirement in the applicable Requirements Document.

VERIFICATION REPORT GUIDELINES

SECTION	CONTENTS
Title Page	The title page should follow the Transportation Agency procedures or style guide. As a minimum, it should contain the following information: ▪ VERIFICATION REPORT FOR THE Name of Project AND Transportation Agency ▪ Contract number ▪ Date the document was formally approved ▪ The organization responsible for preparing the document ▪ Internal document control number, if available ▪ Revision version and date issued
1.0 Purpose of Document	This section identifies the type of verification performed. For instance, the activity may verify the entire system, a sub-system, the deployment at a site, a burn-in test, or any other verification activity called for in the Program Plan or in the SEMP. This section can be taken from the applicable Verification Procedure.
2.0 Identification of the Configuration under test	This section identifies the equipment and software verified. It also identifies all equipment and software necessary for this verification activity that is external to the system / sub-system configuration under test. This may include special test equipment and any external systems with an interface to the configuration under test. This section can be taken from the applicable Verification Procedure.
3.0 Individual Test Case Report	This section summarizes the purpose and results of each test case performed in the applicable Verification Procedure. Special attention is paid to any test case where a failure occurred and how the failure was resolved. This section covers: ▪ Test case overview and results ▪ Completed Verification Procedure pages annotated with pass / fail results ▪ Description of each failure, if any, from the expected result called for in the Verification Procedure ▪ Any back-up data or records related to the field procedure ▪ Details of the resolution of each test failure, including procedure modification, software fix, re-testing and results, regression testing and results, required document changes (including changes to the requirements).

7.5.10 Deployment Plan Guidelines

Purpose of this Document

Deployment is the final step in development of a system. A Deployment Plan is developed based on a thorough analysis of the steps necessary to achieve the deployment goals of the project. It both serves to justify the strategy for deployment and to inform all deployment participants (and other stakeholders) of what will happen and what they will be required to do.

These two parts of the plan serve different purposes and should be written at different times. The strategy section shows management (and the operations people who will get the system) what the selected strategy is and how it best meets the constraints placed on the project (for instance, a multi-year funding profile and viable operational capabilities at each step).

The plan section is just that, a detailed plan for each deployment step, answering what, when, where, how and by whom. This part is best written when the design is fairly complete and the exact system components, and their characteristics, are known in great detail.

Tailoring this Document to Your Project

You may not need a separate written Deployment Plan at all, especially if all deployment takes place at one location and at one time. On the other hand, if there are multiple locations, multiple deployments at each location and many external interfaces (other systems) involved, a Deployment Plan can be very helpful.

It is also possible that only one of the two parts of the Deployment Plan (as mentioned above) is needed. Specifically the time spent in preparing the strategy section very much depends on how much "selling" of the plan is needed.

Project management may also decide that the subject of deployment is covered well enough in other documents (especially the Project Plan, the SEMP and the Verification Plan as well as installation and construction drawings) that a separate Deployment Plan Document is not necessary. There are many factors to be considered, but the most important is, can the deployment be successful without the expense of developing a Deployment Plan?

Checklist of Critical Information

- ☑ Are all the important, and significant, deployment goals and objectives captured?
- ☑ Have as many as possible of the viable deployment strategies been analyzed and compared?
- ☑ Are the strengths of the recommended deployment strategy fully explained?
- ☑ Does the recommended deployment strategy include a clear description of the operational capabilities that exist after each deployment step?
- ☑ Has the recommended deployment strategy been presented to the appropriate stakeholder decision makers?
- ☑ Has the recommended deployment strategy been accepted by the stakeholder decision makers?
- ☑ Are all of the deployment phases included in the Deployment Plan?
- ☑ Are all of the prerequisites to starting each deployment step included and is the responsible party for each identified?
- ☑ Are the installation plans needed for each deployment step identified?
- ☑ Is the list of hardware and software products needed for each deployment step identified?
- ☑ For each deployment, are all participants identified?

DEPLOYMENT PLAN GUIDELINES

SECTION	CONTENTS
Title Page	The title page should follow the Transportation Agency procedures or style guide. As a minimum, it should contain the following information: ▪ DEPLOYMENT PLAN FOR THE Name of Project AND Transportation Agency ▪ Contract number ▪ Date the document was formally approved ▪ The organization responsible for preparing the document ▪ Internal document control number, if available ▪ Revision version and date issued
1.0 Purpose of Document	A brief statement of the purpose of this document, that is, the plan for deploying the systems of the project over one or more phases and into one or more physical locations (sites).
2.0 Scope of Project	This section gives a brief description of the planned project and the purpose of the system to be built. Special emphasis is placed on the project's deployment complexities and challenges. This section may be lifted from earlier documents. It is important only to people (stakeholders) who will be introduced to the project for the first time by this document.
3.0 Deployment 4.0 Strategy	A complex deployment, involving multiple deployment steps at multiple sites, is based on certain goals and objectives. This section lists those goals and objectives and is used to "sell" the Deployment Plan to the stakeholders. It is also important that the deployment participants understand why the deployment is proceeding as it is so they can work with and support the plan. The significant goals and objectives guiding the deployment strategy should be relatively few (no more than a dozen) and need to be clearly stated in this section. Some typical examples of goals and objectives include: ▪ The funding profile for a multi-year project which limits the scope of deployment in a single year ▪ Development and installation prerequisites. An analysis of the system may show that feature A must be deployed first before features B, C or D, all of which need A to function. ▪ Construction activities that must precede deployment. ▪ Deployment of interfacing systems (especially by other agencies) that must precede deployment of a system feature ▪ The need to create a viable operational capability at each stage of the deployment. This influences how much of the system must be deployed at each step. Following the statement of the goals and objectives, a high level view of the deployment strategy is presented. This covers and describes each phase of deployment at each of the sites involved. It describes what is deployed, where it is deployed, and what operational capabilities are the results of this phase of the deployment. It ties the plan to the previously identified goals and objectives so that the stakeholders can understand the rationale for each phase. This summary should include an estimate of the cost of each phase to show that the plan satisfies the funding profile and should show the overall deployment schedule.

SECTION	CONTENTS
5.0 Phase 1 Deployment	This, and the following sections, define and explain each phase of the deployment. The intent here is to identify all the needed participants and to describe to them what they have to do. As will be seen in the following list of section contents, not only are the deliverable products identified, but so are any site work that must be done prior to installation, as well as all activities necessary to show that the deployment was successful and the system is ready for operations, or what ever comes next. In general, each phase description should identify: ▪ The location of the deployment activities ▪ The project-developed equipment and software products to be deployed. Initially this is just a high level list but eventually the list must be exact and complete, showing part numbers and quantity. If detailed hardware installation drawings have been prepared, they are referenced here. ▪ All site work (including construction and facilities) that is needed before installation can begin. Again, reference to drawings may be required. Also any necessary inspection and testing of this work is defined. ▪ All integration activities that need to be performed after installation, including integration with on-site systems and with external systems at other sites. ▪ All verification activities (as defined in the applicable Verification Plan) that must occur prior to acceptance of the site. ▪ All supporting activities that must be completed before site acceptance, such as training and manuals. ▪ The responsible parties for each activity ▪ The schedule for each activity
5.0 Phase N Deployment	This, and any needed additional sections, follows the format for section 4. Each covers a phase of the deployment effort.

7.5.11 Operation and Maintenance Plan Guidelines

Purpose of this Document

This document describes how the finished system will be operated and maintained. Operation and maintenance activities were described in Section 4.6.2. These guidelines describe the scope and content of the Operation and Maintenance Plan, which covers both hardware and software.

The Operation and Maintenance Plan is prepared incrementally during system implementation, and revised as needed during on-going system operation. The first version should be produced as early in the project as possible, to ensure that operation and maintenance needs are understood and planned for. This initial version may be quite limited in content, focusing on issues such as staffing, funding, and documentation that need to be worked on well in advance of system startup. Details of specific operation and maintenance activities can be added as needed, and after the system is developed and its specific characteristics are known.

The Operation and Maintenance Plan is separate from operating manuals and maintenance manuals provided by system or component developers or suppliers. Those documents describe detailed procedures, whereas the O&M Plan describes resource organization, responsibilities, policies, and general procedures. For example, the O&M Plan may say that the system administrator will ensure that databases are backed up daily. An operation or maintenance manual will describe how to do a backup.

Tailoring this Document to Your Project

Operation and maintenance activities can usually be described in a single plan, but for large or complex systems it may be appropriate to prepare a maintenance plan separately from the operation plan. Similarly, large or complex systems may warrant separate plans for specific aspects of operation or maintenance, including for example, configuration management, staff training, data management, safety and security.

Some sections of the document described below may not be needed for a particular system. Other systems may need additional sections not mentioned here. The plan should provide sufficient information for the system to be effectively operated and maintained, even in the event of a complete turn-over of the personnel originally involved.

The project Concept of Operations, System Requirements, and Design Documents will provide initial guidance as to the extent and nature of operation and maintenance activities. As specific components are procured and implemented, the Plan can be updated and expanded to include more specific information.

For small or simple systems, configuration management may be covered within the Operation and Maintenance Plan, but otherwise will be the subject of a separate plan (see 7.5 Configuration Management Plan). The two are closely related.

Since the Operation and Maintenance Plan needs to be used and updated throughout the life of the system, it is not appropriate to merely make it a section within the Project Plan.

Checklist of Critical Information

- ☑ Does the Operation and Maintenance Plan answer all the questions of who, what, where, and when concerning operation and maintenance?
- ☑ Does the Plan identify the personnel responsible for operation and maintenance?
- ☑ Does the Plan identify the human resources and facilities, including tools, needed for operation and maintenance?
- ☑ Does the Plan identify funding sources for on-going operation and maintenance?
- ☑ Does the Plan describe the operation and maintenance activities to be performed?
- ☑ Does the Plan describe the checks to be made, and the data to be collected, for health and performance monitoring?
- ☑ Does the Plan cover periodic reporting of system health and performance to provide feedback to management on the effectiveness of operations and maintenance?
- ☑ Does the Plan address the training of operators and maintenance personnel?
- ☑ Does the Plan address safety and security?
- ☑ Does the Plan identify other documents used in operations and maintenance, such as relevant policy directives, system configuration documentation, and operating and maintenance manuals?
- ☑ Does the Plan address system testing, and configuration documentation updates (may be dealt with in a separate Configuration Management Plan), following configuration changes, repairs, and upgrades?

- ☑ Does the Plan address preventive maintenance as well as reactive maintenance?
- ☑ Does the Plan address expected life and end-of-life replacement or upgrade?

OPERATION AND MAINTENANCE PLAN GUIDELINES

The following format is only one example of many alternatives. If the new system is just one of multiple systems operated and maintained by the same personnel, the material described here may be incorporated in an existing Operations and Maintenance Plan covering multiple systems.

<table>
<tr><th>SECTION</th><th>CONTENTS</th></tr>
<tr><td>Title Page</td><td>The title page should follow the Transportation Agency procedures or style guide. As a minimum, it should contain the following information:
OPERATION AND MAINTENANCE PLAN FOR THE Name of System
The organization responsible for preparing the document
Internal document control number, if available
Revision version and date issued</td></tr>
<tr><td>1.0 Purpose of Document</td><td>This section identifies the scope and purpose of the Plan, and explains how it fits in with related documents such as the Configuration Management Plan operating manuals, and maintenance manuals. Included is a brief description of the system being operated and maintained, together with its stakeholders, such as agencies and departments within agencies that rely on its successful operation. The system description should list all the system elements that are the subject of this document, including for example, auxiliary equipment and facilities such as any special air conditioning, communications links, special lighting, and/or special furniture.</td></tr>
<tr><td>2.0 Facilities and Resources</td><td>This section identifies the facilities and resources to be used for system operation and maintenance. It should cover at least the following elements:
Personnel, including positions, general qualifications, specialty skills needed, and percentage of time dedicated to system operation or maintenance if not full time.
Building space, including for example, rooms, space within rooms, specialty areas, such as workshops, raised floors, additional air conditioning, additional power, communications trunks.
Furniture, equipment, and tools.
Training needed for operations and maintenance personnel, including off-site courses, on-site courses, and hands-on training on the system itself.
Funding, including the amount needed each year and sources. Attempt to predict future costs, including unusual items such as end-of-life replacement.
Clearly identify the demarcation between facilities, resources, and funding that are treated as system-specific versus those that are already in place and general purpose and assumed to remain so..</td></tr>
<tr><td>3.0 Operations</td><td>This section describes policies and high-level procedures governing operation of the system. As a minimum, it should address the activities described in the project’s Concept of Operations, and any other activities needed to achieve the project’s objectives.
In general, the following information should be included in this section:
A clear statement of system operation goals and expectations.</td></tr>
</table>

SECTION	CONTENTS
	▪ Hours of operation (if not continuous) or the conditions that trigger commencement and termination of intermittent system operation. ▪ Automated processes involved in system operation. ▪ Operation activities (including monitoring of automated processes) needing human involvement and the personnel responsible for each. ▪ Backup facilities and personnel, and procedures for invoking use of backups. ▪ Interaction and coordination needed with other systems and personnel, including policies for decision making, overrides, and notification in the event of competing interests. ▪ Special procedures and interactions that apply in the event of major emergencies. ▪ Parameters used to monitor the effectiveness of system operation, and how those data are to be collected and reported. ▪ Policies covering security, including for example, physical access, system access (e.g., log in/out, password management, remote access, and firewalls.), and fire and safety. ▪ Procedures related to system health monitoring and reporting, initiation of maintenance actions, and hand-off between operation and maintenance personnel at both the start and end of maintenance actions. ▪ Policies regarding data collection and archiving, including what data are to be stored for how long. ▪ Policies regarding privacy, such as restrictions on the use of cameras and recording of information that may be able to identify individuals. ▪ Policies regarding visits, telephone enquiries, and other interactions with interested parties such as other ITS professionals, researchers, news reporters, and the public.. ▪ Construction activities that must precede deployment. ▪ Deployment of interfacing systems (especially by other agencies) that must precede deployment of a system feature ▪ The need to create a viable operational capability at each stage of the deployment. This influences how much of the system must be deployed at each step. Following the statement of the goals and objectives, a high level view of the deployment strategy is presented. This covers and describes each phase of deployment at each of the sites involved. It describes what is deployed, where it is deployed, and what operational capabilities are the results of this phase of the deployment. It ties the plan to the previously identified goals and objectives so that the stakeholders can understand the rationale for each phase. This summary should include an estimate of the cost of each phase to show that the plan satisfies the funding profile and should show the overall deployment schedule.
4.0 Maintenance	This section describes policies and high-level procedures governing maintenance of the system. It should address both proactive (preventive) and reactive (corrective) activities needed to keep the system fully operational.

<table>
<tr><th>SECTION</th><th>CONTENTS</th></tr>
<tr><td></td><td>In general, the following information should be included in this section:
▪ Preventive maintenance activities and the time schedule or other triggers for each activity.
▪ Corrective maintenance activities, the relative urgency of each, and the maximum target response and correction times for each type of fault.
▪ Policies with regard to purchase of spare equipment, manufacturer or vendor maintenance agreements or extended warranties, and third party maintenance contracts.
▪ Parameters used to monitor the effectiveness of system maintenance, and how those data are to be collected and reported.
▪ Procedures for coordination with operations personnel and activities
▪ Demarcation of responsibilities relative to maintenance by other parties and procedures for coordination with personnel responsible for interconnected systems or components that are not part of this system.</td></tr>
<tr><td>Appendix</td><td>A list of the names and contact information of personnel currently assigned to system operation and maintenance. Include the names and contact information of personnel in other parts of the organization or in other organizations, including emergency response services, with which system operations and maintenance personnel must interact</td></tr>
</table>